AIR FORCE INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY
UNITED STATES AIR FORCE

SCHOOL OF ENGINEERING

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

HUFF, A ONE-DIMENSIONAL HYDRODYNAMICS CODE FOR STRONG SHOCKS

THESIS

AFIT/GNE/PH/78D-9

David J. Peters
Captain USAF

(See 1473)

Approved for public release; distribution unlimited

AFIT/GNE/PH/78D-9

HUFF, A ONE-DIMENSIONAL
HYDRODYNAMICS CODE FOR
STRONG SHOCKS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

David J. Peters
Captain USAF
Graduate Engineering Physics
December 1978

Approved for public release; distribution unlimited.

Preface

This report contains the results of my efforts to develop and describe a relatively fast and reasonably accurate computer code designed to simulate time-dependent, high-pressure shock propagation through a gas or solid. The most important result of my work is the one-dimensional hydrodynamic computer code HUFF that I developed. I have tried to present a complete description of the derivation of the model, the assumptions made, and the steps necessary to define and solve problems using the code here at AFIT. I made an extensive effort to provide a simple and flexible program so that HUFF could be used in whole or in part in solving related problems in the future.

There are many people to whom I am indebted and would like to thank. First, I would like to thank Captain Clarence Anderson for providing me with numerous references and extensive background to begin my work. I would also like to thank my advisor, Major George H. Nickel, for his help, guidance, and encouragement during the research period. Last, but most important, I would like to thank my wife Martha for her patience, love, and understanding throughout the period of this work.

David J. Peters

Contents

Page

List of Figures

Figure Page

List of Tables

Abstract

HUFF is a one-dimensional Lagrangian hydrodynamics computer code developed from the basic principles of mass, momentum, and energy conservation for strong shock propagation in a solid or gas. Two equations of state are used - the adiabatic ideal gas law with a variable gamma and the Gruneisen solid equation of state with a constant Gruneisen ratio. The Richtmyer and Morton difference equations for strong shocks are used on a spatial mesh composed of up to 100 cells. Results for two problems are presented which show the usefulness and limitations of the code and also serve as sample problems. The results of a one kiloton nuclear explosion are compared to the Nuclear Blast Standard (1KT). The results were within 13 percent for shock overpressure and overdensity, 5 percent for shock material velocity, and 2 percent for shock position over a range of 20 meters to 2 kilometers from the burst point. The larger deviations occurred at early times being attributed to an absence of radiation transport calculations in the code. The second problem, a megabar compression of uranium, shows agreement within two percent for all parameters (peak shock pressure, density, material velocity and shock velocity) when compared with the Rankine-Hugoniot compression curves. The equation of state for a solid was limited to calculations below 100 megabars due to its simplicity and constant value for the Gruneisen ratio. A complete users guide and program listing are also provided.

HUFF, A ONE-DIMENSIONAL HYDRODYNAMICS CODE FOR STRONG SHOCKS

I. Introduction

One of the primary effects of low altitude nuclear weapon detonations is the development and propagation of strong shock waves. Large computer codes have been developed to accurately describe the development and propagation of shock waves in air and in solids (Ref 10:1). These computer codes generally require large computer storage and long computation times.

In 1975, Clarence L. Anderson developed a computer program based on conservation of mass, momentum, and energy to calculate hydrodynamic shock propagation in solids and in air. Anderson's program used a course spatial mesh (100 cells) with a rezoning capability and some simplifying assumptions to develop a less accurate but more simple operating code (Ref 1:1). The results of Anderson's work was encouraging and lead to the development of HUFF.

The primary purpose of this thesis was to develop a simple, stable school use hydrodynamic computer code based on conservation principles and including appropriate simplifying approximations providing reasonably accurate results with relatively short computational time requirements.

This report presents the conservation principles on which the computer code is based and the application of these principles to the computer. Development of the computer code is discussed and the applicable derivations are presented. Hydrodynamic shock propagation in air and in solids is developed and sample problems are presented.

The development of the hydrodynamic shock code is presented in Chapter II. Chapter III discusses the initial conditions and boundary conditions applicable to the HUFF program and their coupling with the HYDRO subroutine. Chapter IV discusses the results of two sample problems and Chapter V contains the conclusions of this study.

II. Hydrodynamic Shock Propagation

This chapter presents the basic differential equations that govern the propagation of strong shocks in both gases and solids. Various simplifying assumptions are made to reduce the equations to a one-dimensional form. The Lagrangian coordinate system and subscript notation used in HUFF are presented followed by the finite difference scheme used to approximate the governing equations. The stability conditions that constrain the size of the time step and the variable gamma modifications complete the chapter.

Basic Governing Equations

The basic equations governing strong shock propagation are derived from the physical laws requiring that mass, momentum, and total energy be conserved. For propagation of strong shocks in solids the hydrostatic pressures are very large and the solid can be treated as a fluid. Thus the differential equations expressing mass, momentum, and total energy conservation apply equally to both gases and solids (Ref 5:7). The equations are:

$$\frac{\partial \rho}{\partial t} + (\vec{u} \cdot \nabla) \rho + \rho \nabla \cdot \vec{u} = 0 \qquad (1)$$

$$\frac{\partial u}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} + \frac{1}{\rho} \nabla P = 0 \qquad (2)$$

$$\frac{\partial E}{\partial t} + (\vec{u} \cdot \nabla) E + \frac{1}{\rho} \nabla \cdot (P\vec{u}) = 0 \qquad (3)$$

where ρ is density, $\vec{u}$ is velocity, t is time, P is pressure and E is energy.

The equation of state for the solid or gas is used to couple the differential equations and provide a means of solution. The equation of state expressed in its most general form is (Ref 8:294):

$$I = f(P,V) = E - \frac{u^2}{2} \tag{4}$$

where I is internal energy and

$$V = \frac{1}{\rho} \tag{5}$$

Assumptions and Resulting Equations

Simplifying assumptions are necessary to reduce the multi-dimensional equations describing shock propagation to the less general one-dimensional equations solved in HUFF. The assumptions are as follows:

1. The local fluid velocity is directed along and depends on the Eulerian one-dimensional coordinate.
2. Adiabatic conditions exist throughout the problem.
3. An artificial positive viscosity, Ω , can be introduced to spread out the density and pressure discontinuities that develop at the shock front due to the adiabatic assumption.
4. The problem is assumed to occur in free space with no gravity dependence.

The differential equations in one-dimensional form for conservation of mass, momentum, and energy are:

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + \rho\,\frac{\partial u}{\partial x} + \frac{(L - 1)u}{x} = 0 \qquad (6)$$

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{1}{\rho}\,\frac{\partial P}{\partial x} = 0 \qquad (7)$$

$$\frac{\partial E}{\partial t} + u\,\frac{\partial E}{\partial x} + \frac{1}{\rho}\,\frac{\partial Pu}{\partial x} + \frac{(L - 1)Pu}{x} = 0 \qquad (8)$$

where L is the geometry factor (L = 1 for rectangular, L = 2 for cylindrical, and L = 3 for spherical).

Langrangian Coordinate System

The spatial mesh used in HUFF is constrained to conserve mass between the mesh points. The volume between two mesh points is defined to be a cell. Depending on the geometry, the mesh points define cell boundaries that are either infinite planes, infinite concentric cylinders, or concentric spheres. Each cell contains a fixed mass for all time - except during rezoning in the blast problem where cells are redefined by removing every other intermediate cell boundary. Requiring a cell to contain a fixed mass defines what is called a Lagrangian mesh. The position of the cell boundaries are followed in time and no fluid moves through the cell boundaries (Ref 5:82).

In a Lagrangian coordinate system the mass of each cell remains fixed. As time progresses, the cell boundary position changes. The density of a cell at time, t , is given by

$$\rho(x) = \rho_0(x_0) \left(\frac{\partial x_0}{\partial x}\right)_t \left(\frac{x}{x_0}\right)^{L-1} \tag{9}$$

where x_0 and ρ_0 are the original position and density respectively. The velocity of a cell boundary is given by

$$u = \frac{\partial x}{\partial t} \tag{10}$$

The Lagrangian equations for one-dimensional shock propagation expressing conservation of mass, momentum, and energy are

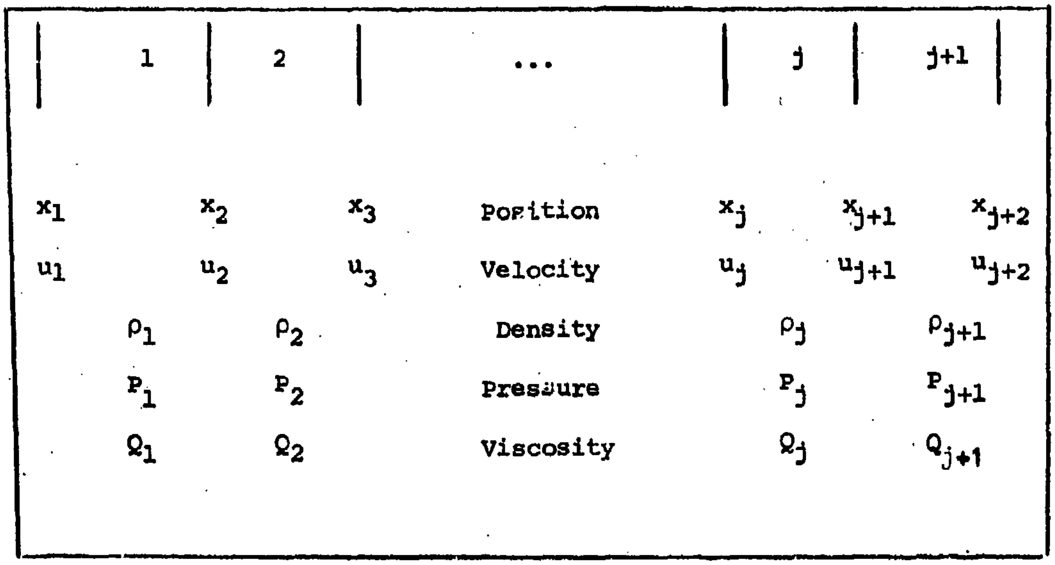
$$\frac{\partial \rho}{\partial t} + \frac{\rho^2}{\rho_0} \frac{\partial u}{\partial x_0} + \frac{(L - 1)Pu}{x} = 0 \tag{11}$$

$$\rho_0 \frac{\partial u}{\partial t} + \frac{\partial P}{\partial x_0} \left(\frac{x}{x_0}\right)^{L-1} = 0 \tag{12}$$

$$\frac{\partial I}{\partial t} + \frac{P\partial V}{\partial t} = 0 \tag{13}$$

Lagrangian Mesh

The Lagrangian mesh material and cell boundary properties are depicted on Figure 1. The cell boundary positions and velocities begin with x_1 and u_1 at the left wall of the mesh. The material properties are cell centered values with ρ_i , P_i , and Q_i located



	1		2		...		j		j+1	
x_1		x_2		x_3	Position	x_j		x_{j+1}		x_{j+2}
u_1		u_2		u_3	Velocity	u_j		u_{j+1}		u_{j+2}
	ρ_1		ρ_2		Density		ρ_j		ρ_{j+1}	
	P_1		P_2		Pressure		P_j		P_{j+1}	
	Q_1		Q_2		Viscosity		Q_j		Q_{j+1}	

Figure 1. Labeling Scheme for Lagrangian Cells and Materials Properties

between the cell boundaries at x_i and x_{i+1}. Program HUFF is dimensioned to contain a grid of 101 values (100 cells). x_1 can be the left wall or center of a rectangular geometry problem while it usually represents the center of a spherical or cylindrical geometry.

Finite Difference Scheme

The choice of a properly centered finite difference scheme is critical in developing a stable code to propagate the shock. The finite difference scheme must approximate the solution in a way that strictly conserves mass, momentum, and total energy. Any difference scheme that is properly centered in both time and space can be expressed in a conservative form (Ref 9:283). Inadvertant changes in the material conditions of any cell in the mesh can lead to propagation of unwanted extraneous waves through the mesh.

The difference equations used in HUFF are those developed by Richtmyer and Morton. The equations are

$$\frac{x_j^{n+1} - x_j^{n}}{\Delta t} = u_j^{n+1} \tag{14}$$

$$\frac{u_j^{n+1} - u_j^{n}}{\Delta t} = -\frac{1}{\rho}\,\frac{(P_j^{n} - P_{j-1}^{n} + Q_j^{n} - Q_{j-1}^{n})}{\Delta x}\left(\frac{x_j^{n}}{(x_0)_j}\right)^{L-1} \tag{15}$$

$$V_j^{n+1} = \frac{1}{\rho_0}\left[\frac{(x_{j+1}^{n+1})^{L} - (x_j^{n+1})^{L}}{\{(x_0)_{j+1}\}^{L} - \{(x_0)_j\}^{L}}\right] \tag{16}$$

$$I_j^{n+1} - I_j^{n} + \left[\frac{P_j^{n+1} + P_j^{n}}{2} + Q_j^{n+1}\right](V_j^{n+1} - V_j^{n}) = 0 \tag{17}$$

$$I_j^{n+1} = f(P_j^{n+1}, V_j^{n+1}) \tag{18}$$

$$Q_j^{n+1} = \begin{cases} \dfrac{2a^2}{V_j^{n+1} + V_j^{n}}\,(u_{j+1}^{n+1} - u_j^{n+1}) & \\ 0 & \text{if } (u_{j+1}^{n+1} - u_j^{n+1}) \geq 0 \end{cases} \tag{19}$$

where I is internal energy, Q is artificial viscosity, and a is the approximate number of cells over which the shock discontinuity is spread (Ref 8:318).

Internal energy is eliminated from the equations by coupling Eq (17) and Eq (18). Equation (18) is the equation of state for the material in the problem. The ideal gas equation of state is used for air and the Gruneisen equation of state for solids.

Gas. The finite difference form of the ideal gas equations of state is (Ref 5:3)

$$P_j^{n+1} = (\gamma - 1)\rho_j^{n+1} I_j^{n+1} \qquad (20)$$

Combining Eq (17) and (20) to eliminate internal energy gives

$$P_j^{n+1} = \frac{P_j^n\left[S(V_j^n) - (V_j^{n+1} - V_j^n)\right] - 2Q_j^{n+1}(V_j^{n+1} - V_j^n)}{S(V_j^{n+1}) + (V_j^{n+1} - V_j^n)} \qquad (21)$$

where γ is gamma and

$$S = \frac{2}{(\gamma - 1)} \qquad (22)$$

A complete derivation of Eq (21) is contained in Appendix D.

Solid. The finite difference form of the Gruneisen equation of state is (Ref 5:3)

$$P_j^{n+1} = (P_H)_j^{n+1} + \gamma_s \rho_j^{n+1}\left[I_j^{n+1} - (I_H)_j^{n+1}\right] \qquad (23)$$

$$(P_H)_j^{n+1} = \frac{c^2(V_o - V_j^{n+1})}{\left[V_o - S(V_o - V_j^{n+1})\right]^2} \qquad (24)$$

$$(I_H)_j^{n+1} = \frac{1}{2}\left[\frac{c(V_0 - V_j^{n+1})}{V_0 - S(V_0 - V_j^{n+1})}\right]^2 \qquad (25)$$

$$S = \frac{(\gamma_s + 1)}{2} \qquad (26)$$

where γ_s is the Gruneisen constant and c is the normal density speed of sound in the material.

Combining Eq (17) and (23) to eliminate internal energy gives

$$P_j^{n+1} = \frac{P_j^n\left[2V_j^n - \gamma_s(V_j^{n+1} - V_j^n)\right] - 2\gamma_s\left[(I_H)_j^{n+1} - (I_H)_j^n\right]}{2V_j^{n+1} + \gamma_s(V_j^{n+1} - V_j^n)}$$

$$+ \frac{2V_j^{n+1}(P_H)_j^{n+1} - 2V_j^n(P_H)_j^n - 2\gamma_s Q_j^{n+1}(V_j^{n+1} - V_j^n)}{2V_j^{n+1} + \gamma_s(V_j^{n+1} - V_j^n)} \qquad (27)$$

A complete derivation of Eq (27) is contained in Appendix D.

The difference equations presented above are solved explicitly in HUFF's Subroutine HYDRO. The solution is examined in more detail in Appendix A.

Stability Conditions

Constraints must be placed on the size of the time step, Δt, to ensure sufficient continuity is maintained in the solution of material properties in the mesh. If the time step is too large, local instabilities will develop overwhelming the solution.

The Lagrangian stability condition provided by Richtmyer and Morton (Ref 8:298) is used to specify the time step in HUFF. The equation is

$$\frac{C_A \Delta t}{X_{j+1}^n - X_j^n} = 1 \tag{28}$$

where C_A is the local adiabatic sound speed (Ref 5:6). The expression for a gas is

$$C_A = \left[\frac{\gamma P_j^n}{\rho_j^n}\right]^{1/2} \tag{29}$$

and for a solid is

$$C_A = \left[\frac{C^2 V_o (V_j^n)^2 (S + 1) - C^2 V_o{}^2 V_j^n S}{\left[V_o - S(V_o - V_j^n)\right]^3} + 2SV_j^n P_j^n\right]^{1/2} \tag{30}$$

where

$$S = \frac{\gamma_S + 1}{2} \tag{31}$$

A complete derivation of Eq (29) and (30) is contained in Appendix D.

An additional stability condition is included in HUFF to ensure the cell boundary positions do not overlap. This condition is

$$(u_{j+1}^{n} - u_{j}^{n}) \Delta t < .5(x_{j+1}^{n} - x_{j}^{n}) \tag{32}$$

This condition becomes a contributing factor when converging cell boundary velocities become very high and density values in the cell have not had time to respond.

The time step is calculated at the beginning of subroutine HYDRO for all the cells in the mesh. The smallest time step value is used for all mesh point calculations during the same iteration. The next iteration begins by computing a new time step.

Variable Gamma Approximation

The actual value for gamma in the gas equation of state is not a constant for real gases. The value ranges from 1.4 under normal atmospheric conditions in air to a value as low as 1.1 under high internal energy conditions. A variable gamma is used in HUFF to calculate the time step and the new pressures in a gas.

A "semi-physical fit" for gamma as a function of internal energy for air was developed by L. R. Doan and G. H. Nickel (Ref 2:5). A rough fit to this function was developed for use in HUFF calculations. The HUFF rough fit gamma function with I expressed in the units 10^{10} ergs/gm is

for $I \leq .1315$

$$(\gamma - 1) = .3981 \tag{33}$$

for $.1315 \leq I \leq 1.0$

$$(\gamma - 1) = -.0399(I - .1315)^2 + .3981 \qquad (34)$$

for $I \geq 1.0$

$$(\gamma - 1) = .16 + \frac{.624}{(2 + I)} \qquad (35)$$

A density correction is included for values of $I \geq 1.0$ and is given by

$$(\gamma - 1) = (\gamma - 1)(\rho/\rho_o)^{\alpha(I)} \qquad (36)$$

where

$$\alpha(I) = .0577\{LOG(I)\} - .0218\{LOG(I)\}^2 + .0035\{LOG(I)\}^3 + .0002\{LOG(I)\}^4 \qquad (37)$$

Figure 2 compares the HUFF fit to the Doan and Nickel equation of state for gamma in air.

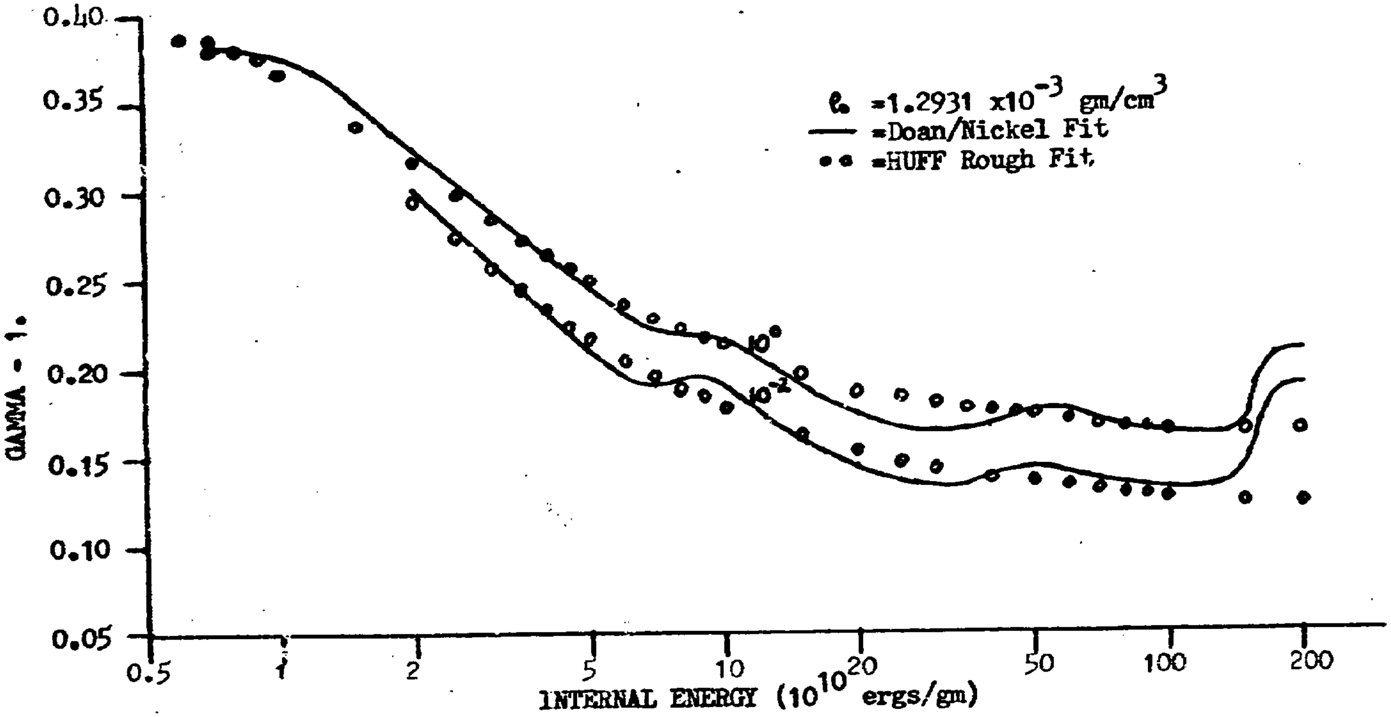


Figure 2. Comparison of HUFF Rough Fit to the Doan and Nickel Equation of State for Gamma in Air

III. Problem Definition and Solution

Program HUFF was developed to be used with a wide variety of one-dimensional problems. Each task accomplished by the computer code is contained in modular units called subroutines. All real or integer values commo' mong subroutines are passed through the subroutine argument list. This type of programming concept is called modular programming (Ref 11).

Program HUFF is designed to solve any one-dimensional single material problem involving hydrodynamic shock propagation in a solid or in air. A spherical blast problem in air and a solid compression problem are included with the program to demonstrate HUFF's capabilities. The spherical blast problem will be discussed first followed by the solid compression problem. A discussion of the flexibility of using HUFF in solving additional hydrodynamic problems is presented last.

Spherical Blast in Air

The spherical blast problem is defined by subroutine BLAST. The initial conditions are based on the following:

1. A large fraction of the yield is absorbed by the air as internal energy.
2. All the absorbed energy is deposited before the air has a chance to respond.
3. The time required to deposite the blast energy is defined as the time to hydroseparation (Ref 4:65).
4. The absorbed blast energy is deposited as internal energy in an isothermal sphere with a radius equal to the fireball

Table I

Average Atmoshperic Parameters for
Mid-Latitudes (Ref 4:104)

Altitude	Scaling Factors			Altitude	Scaling Factors		
(feet)	SP	SD	ST	(feet)	SP	SD	ST
00	1.00	1.00	1.00	15,000	0.56	1.21	1.28
1,000	0.96	1.01	1.02	20,000	0.46	1.30	1.39
2,000	0.93	1.03	1.03	25,000	0.37	1.39	1.53
3,000	0.90	1.04	1.05	30,000	0.30	1.50	1.68
4,000	0.86	1.05	1.07	35,000	0.24	1.62	1.86
5,000	0.83	1.06	1.08	40,000	0.19	1.75	2.02
10,000	0.69	1.13	1.17				

radius at hydroseparation (Ref 4:65).

Subroutine BLAST contains the proper input values for a one kiloton sea level explosion. The subroutine scales the programmed values calculating the proper initial conditions for a variety of blast problems. The yield of the explosive and the altitude parameters provided in Table I are input by the user to define the type of blast problem. The parameters are defined as

$$SP = \frac{P}{P_o} \qquad (38)$$

$$SD = \left(\frac{P_o}{P}\right)^{1/3} \qquad (39)$$

$$ST = (\frac{P_O}{P})^{1/3} (\frac{T_O}{T})^{1/2} \tag{40}$$

where P_o and T_o are normal sea level atmosphere pressure and temperature and P and T are the pressure and temperature at altitude. These parameters (SP, SD, and ST) are used only to define the proper initial conditions for a variety of altitude and yield problems and are not used after the hydrodynamic calculations begin.

Compression of a Solid

The compression problem is defined by subroutine SQUEEZE. An initial constant pressure is applied in the outside cell of the mesh.

The compression problem will compress the solid in any of the three problem geometries (rectangular, cylindrical, spherical). The example problem presented in Chapter IV uses rectangular geometry.

The Gruneisen equation of state used in program HUFF limits the amount of overpressure that can be applied to the solid. The maximum density that can be achieved is less than

$$\frac{P}{P_o} = \frac{\gamma_s + 1}{\gamma_s - 1} \tag{41}$$

At this density, subroutine HYDRO will calculate an infinite value for pressure.

Defining Other Problems

Many additional problems can be defined and solved by program HUFF by defining the initial and boundary conditions in a problem defining subroutine similar to BLAST or SQUEEZE. A general understanding of a few basic concepts used in program HUFF will be helpful when defining a new problem.

The HUFF program flow pattern is depicted in Figure 3. The program begins by calling subroutine FLAGS to establish the "sign posts" that identify the route the program will follow in solving the problem. The main program then transfers complete control of the problem solution to subroutine CONTROL.

Subroutine CONTROL directs the routing of the program during the solution process. The "problem defining subroutine" is called before beginning the solution to establish the initial conditions of the problem. All the remaining task oriented subroutines are called at the proper times during the solution as the program cycles through each time step.

Subroutine HYDRO calculates all the new material properties of the mesh for one time step each time it is called by subroutine CONTROL. The only exception is the two exterior boundary velocities (u_1 and u_{101}). If the boundary velocities are constant throughout the problem solution, no additional adjustments to the HYDRO calculations are required. If the boundary velocities change during the problem, an additional entry from subroutine HYDRO to the "problem defining subroutine" will be necessary to calculate the outer boundary velocities. This call must be accomplished after subroutine HYDRO calculates the time step and before the new cell positions are computed.

The information provided above should be sufficient to allow a programmer to use HUFF to solve a variety of additional problems beyond those provided in the original program. All of the task oriented subroutines contained in HUFF are well-documented to aid the user if further modifications are required.

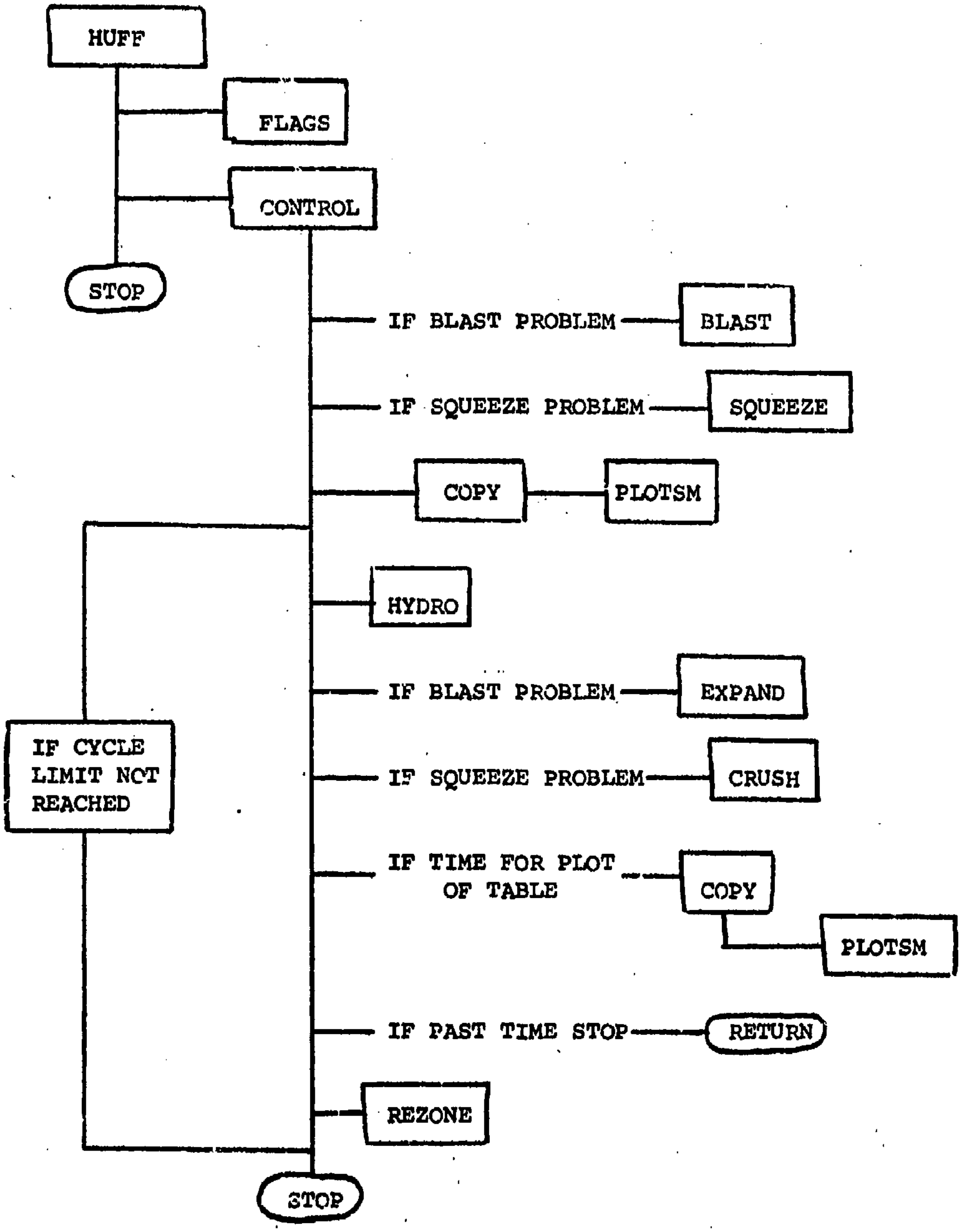


Figure 3. Program HUFF Flow Diagram

IV. Results and Discussion

This chapter, along with Appendices B and C, presents and discusses the results for two sample problems. The first problem discussed is a one-kiloton nuclear burst in infinite sea level air. The second problem is the one-dimensional compression of uranium both in rectangular and spherical geometries. The results of these problems demonstrate the usefulness and limitations of HUFF as an effective first order hydrodynamic computer code.

1 KT Explosion

The one-kiloton nuclear explosion in infinite sea level air was chosen as a sample problem because of the availability of other results to check those of HUFF. The time used in the initial conditions was .0368 milliseconds corresponding to the time of hydroseparation (Ref 4: 65, 103). The radius of the isothermal sphere at hydroseparation is 450 cm. An input overpressure of 1.5 x 10^{10} dynes/cm^2 was used to represent the total energy deposited in the isothermal bubble by the time of hydroseparation. The input pressure corresponds to the total energy deposited in the air by the time of hydroseparation (Ref 4:64). The time calculated by HUFF is the time after the nuclear blast detonation.

The results from HUFF were compared with those from the Nuclear Blast Standard (1KT) (Ref 7) and BEERAY (Ref 1:36). The results compared quite well with the standard showing a significant improvement over the results of BEERAY.

HUFF was compared with the 1KT standard for all the times published in the standard over the time range of three millisecond to two seconds after detonation. HUFF's accuracy deteriorated after two seconds due to its failure in simulating weak shock propagation at great distances. The range during the first two seconds is from 27 meters to 2.6 kilometers.

The shock position calculated by HUFF is within three meters of the standard throughout the range from 27 meters to beyond one kilometer. At distances beyond one kilometer to over three kilometers, it agrees with the standard within two percent. Almost all of this uncertainty can be attributed to the relatively large cells of the mesh (.5 meters per cell at 27 meters and 8 meters per cell at one kilometer).

The maximum shock pressure agrees generally within seven percent of the standard with a maximum error of 13 percent occurring around 50 milliseconds. The pressure is generally higher at early times than the standard and lower than the standard at late times. HUFF's deviation from the standard could be the result of neglecting to compensate for the energy lost and gained in the air by radiation from the fireball and the early shock. A significant percentage of the total weapon yield is carried away by radiation and deposited ahead of the shock. This energy is absorbed by the air changing the normal conditions assumed by HUFF. Both pressure and density are sensitive to this additional energy transfer.

The maximum shock density agrees generally within five percent of the standard. The largest deviation was 12 percent occurring around 50 milliseconds corresponding to the high pressure present at this time. Much of this additional energy represented by a high pressure and density would be converted to radiation energy and transmitted ahead of the shock due to the high temperatures present. HUFF does not contain a thermal radiation code to handle this condition properly. At later times, pressure and density both return to closer agreement with the standard and fall below the standard at great distances. This agrees with what would be expucted from a code neglecting radiation transport.

The maximum material velocity at the shock agrees within six percent of the standard throughout the strong shock propagation phase. This covers a range from 27 meters to over two kilometers. The material velocity remains slightly above the standard throughout the problem.

In general, HUFF agrees with the Nuclear Blast Standard peak values within 10 percent with few exceptions. This is a remarkable correlation considering the small spatial mesh used (nine cells initially and 50-100 cells at later times) and the absence of a subroutine to handle thermal radiation. Tables II and III give a comparison of HUFF's results for three milliseconds and one second respectively.

HUFF has also maintained the efficiency of BEERAY even though additional steps were added to improve HUFF's accuracy. The addition of a variable gamma in the equation of state for air significantly increased the computational time required to compute each time step (cycle). This additional computational time per cycle was offset by the incorporation of a "hydromatic" time step control which reduced the number of cycles required to reach two seconds from 9000 iterations in BEERAY to 2620 in HUFF. Figure 4 through 6 further display HUFF's performance.

Compression of Uranium

The rectangular geometry compression problem was selected to test the equation of state and the solution process used in HUFF in solving problems involving solids. The material selected for the problem

Table II

Comparison of the Results of HUFF with
the One Kiloton Standard

Time = 3.00×10^{-3} sec (471 Time Iterations for HUFF)

	1KT Standard	HUFF	%Difference
Shock Position (cm)	2.9×10^{3}	2.7×10^{3}	-7
Shock Overpressure (dynes/cm^2)	1.45×10^{8}	1.52×10^{8}	6
Shock Overdensity (gm/cm^3)	7.50×10^{-3}	7.44×10^{-3}	-1
Shock Material Velocity (cm/sec)	3.45×10^{5}	3.38×10^{5}	-2

(Ref 7:22-26)

Table III

Comparison of the Results of HUFF with
the One Kiloton Standard

Time = 1.000 sec (2340 Time Iterations for HUFF)

	1KT Standard	HUFF	%Difference
Shock Position (cm)	5.04×10^{4}	5.05×10^{4}	0
Shock Overpressure (dynes/cm^2)	2.3×10^{4}	2.2×10^{5}	-4
Shock Overdensity (gm/cm^3)	2.0×10^{-4}	1.94×10^{-4}	-3
Shock Material Velocity (cm/sec)	5.09×10^{3}	5.04×10^{3}	-1

(Ref 7:74-81)

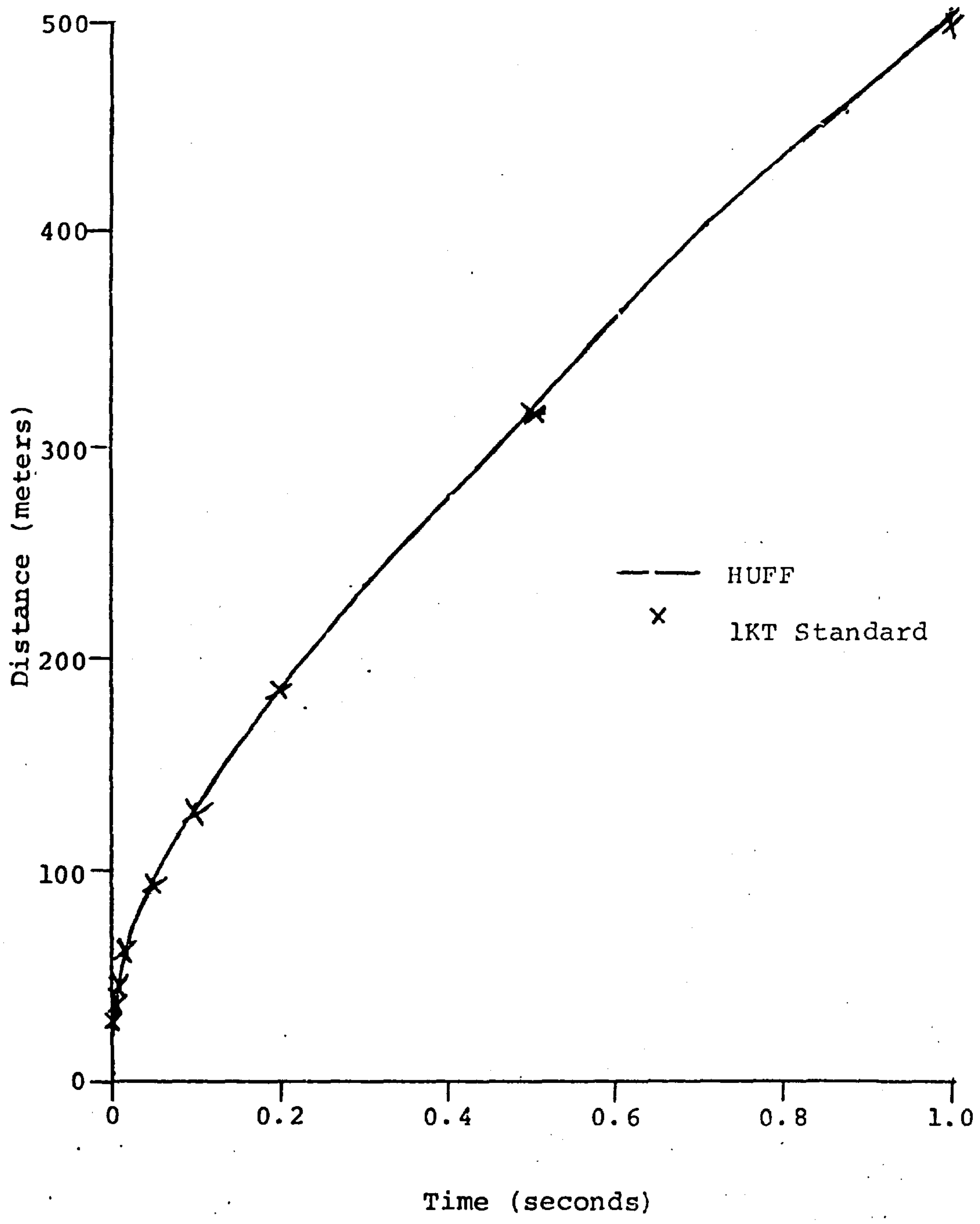


Figure 4. Comparison of HUFF's Ground Range vs Time with the 1KT Standard

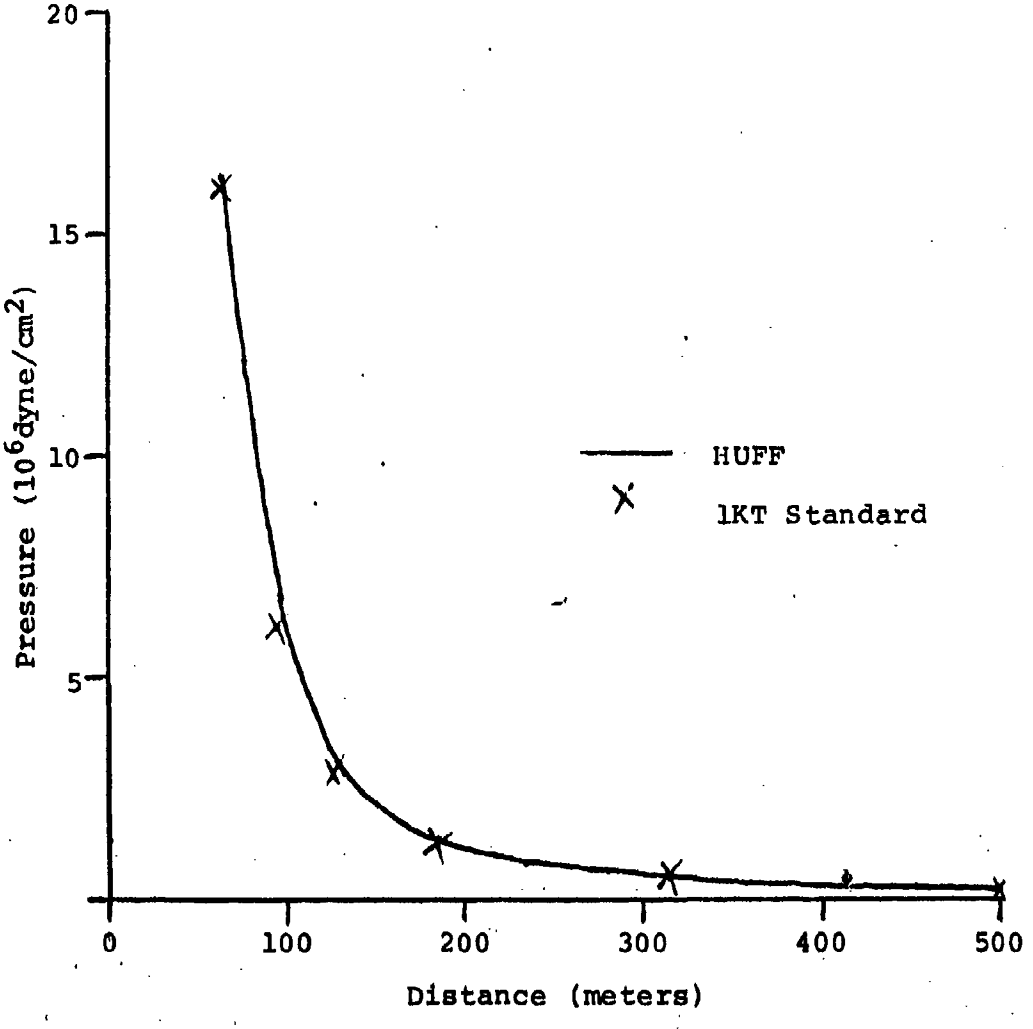


Figure 5. Comparison of HUFF's Overpressure vs Radius with the 1KT Standard

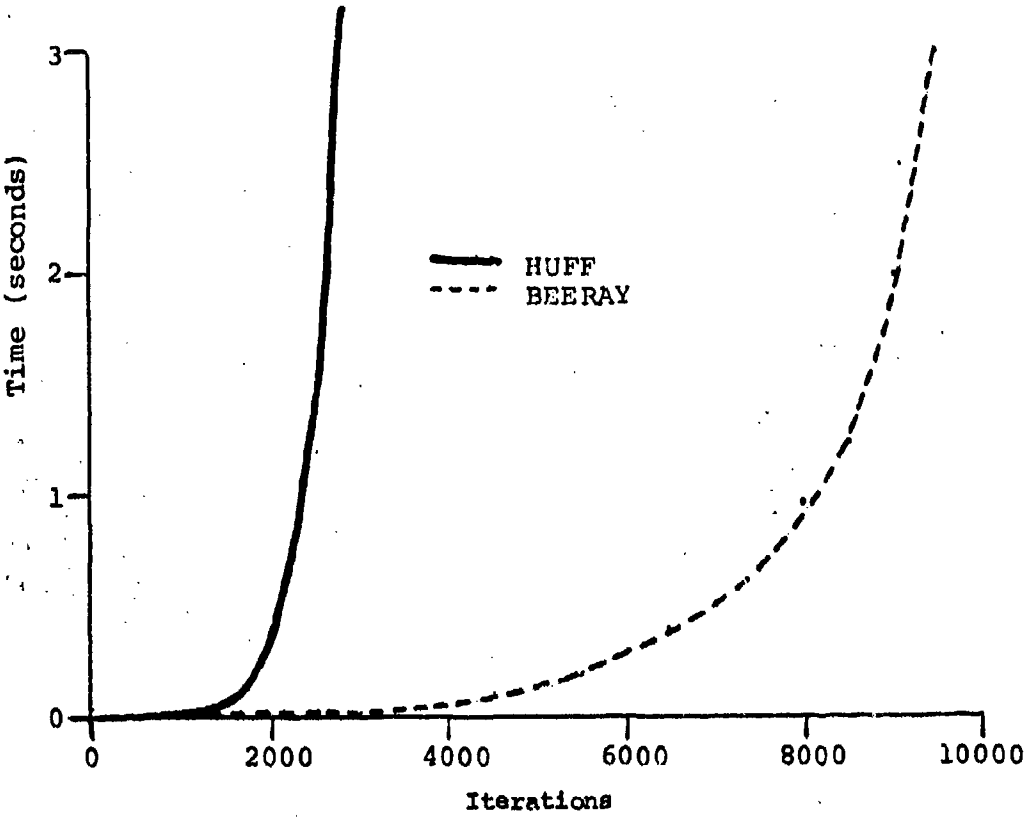


Figure 6. Comparison of HUFF's Time vs. Iterations with BEERAY in the Blast Problem (total execution time for both HUFF and BEERAY are about equal due to the additional time per iteration required by HUFF to include the variable gamma computations)

was uranium with three percent molyldenum by weight for which well documented data is available on the Rankine-Hugoniot parameters (Ref 6: 534). This same material was used in BEERAY providing an additional comparison for HUFF's results.

Appendix C contains the input data and some of HUFF's plotted results for this problem. Figure 7 shows the peak shock pressure vs. time and Figure 8 shows the peak density vs. time. A constant pressure of one megabar was applied to the outer cell of the mesh propagating a shock inward. Figure 9 shows a plot of position vs. time for the shock.

Table IV contains the results of HUFF compared to the Rankine-Hugoniot parameters and BEERAY's results. The values given for HUFF are an average of 15 data points taken from a time range from 2×10^{-6} to 2×10^{-5} seconds. This corresponds to a shock position range from 9 to 1.5 cm respectively in the 10 cm initial radius sphere of uranium.

The shock pressure at the shock front is slightly higher than the pressure behind the shock. This results in a slightly higher density and material velocity at the shock front. A two percent increase in pressure at this point on the Hugoniot curves corresponds to a density increase of about one percent due to the slope of the curve at this point (Ref 2:87).

Appendix H contains the plotted results for the HUFF compression problem in a spherical geometry. In this geometry, the parameters change as the shock travels toward the center of the sphere. Comparing the density and velocities at the point in the sphere where the pressure is one megabar gives similar results to that observed in the rectangular geometry.

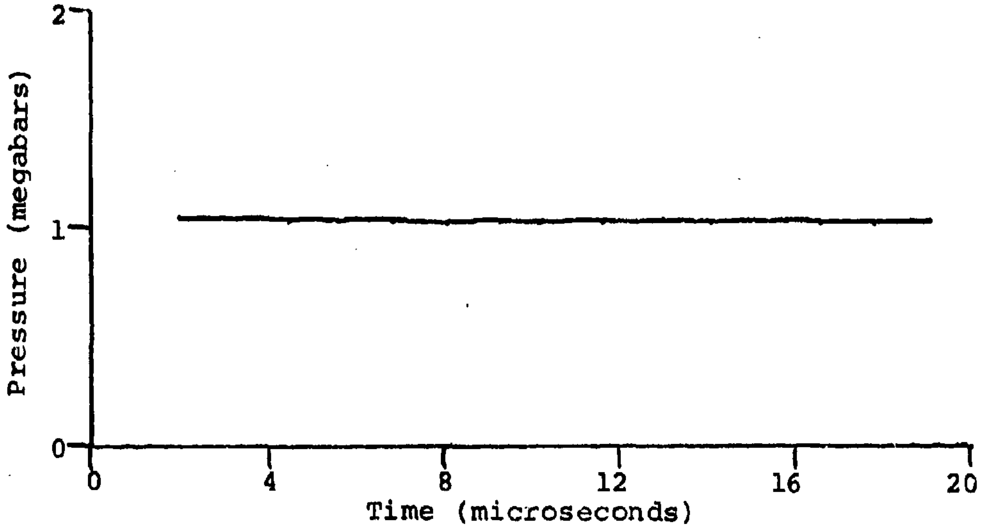


Figure 7. Peak Shock Pressure vs Time in the Rectangular Compression of Uranium

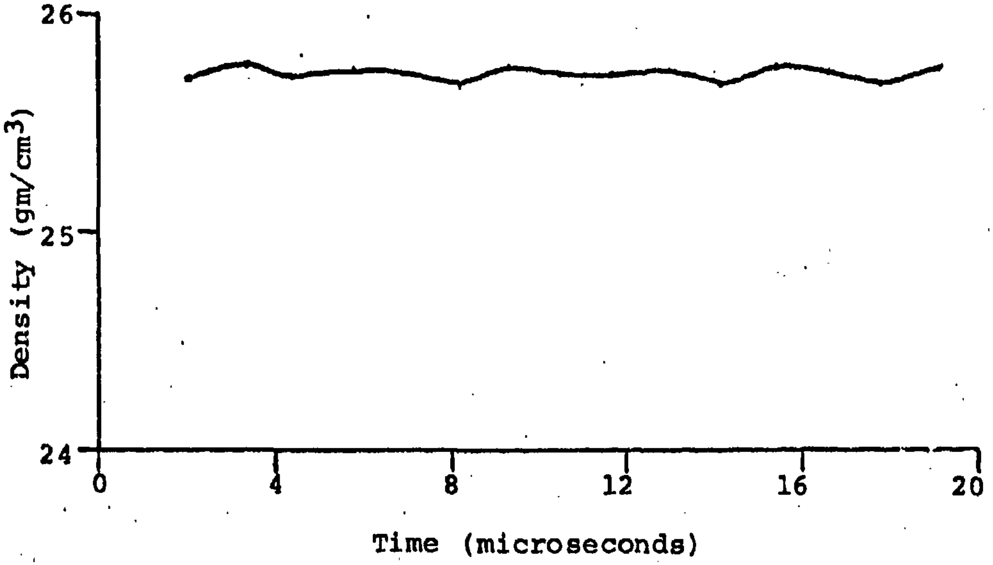


Figure 8. Peak Shock Density vs Time in the Rectangular Compression of Uranium

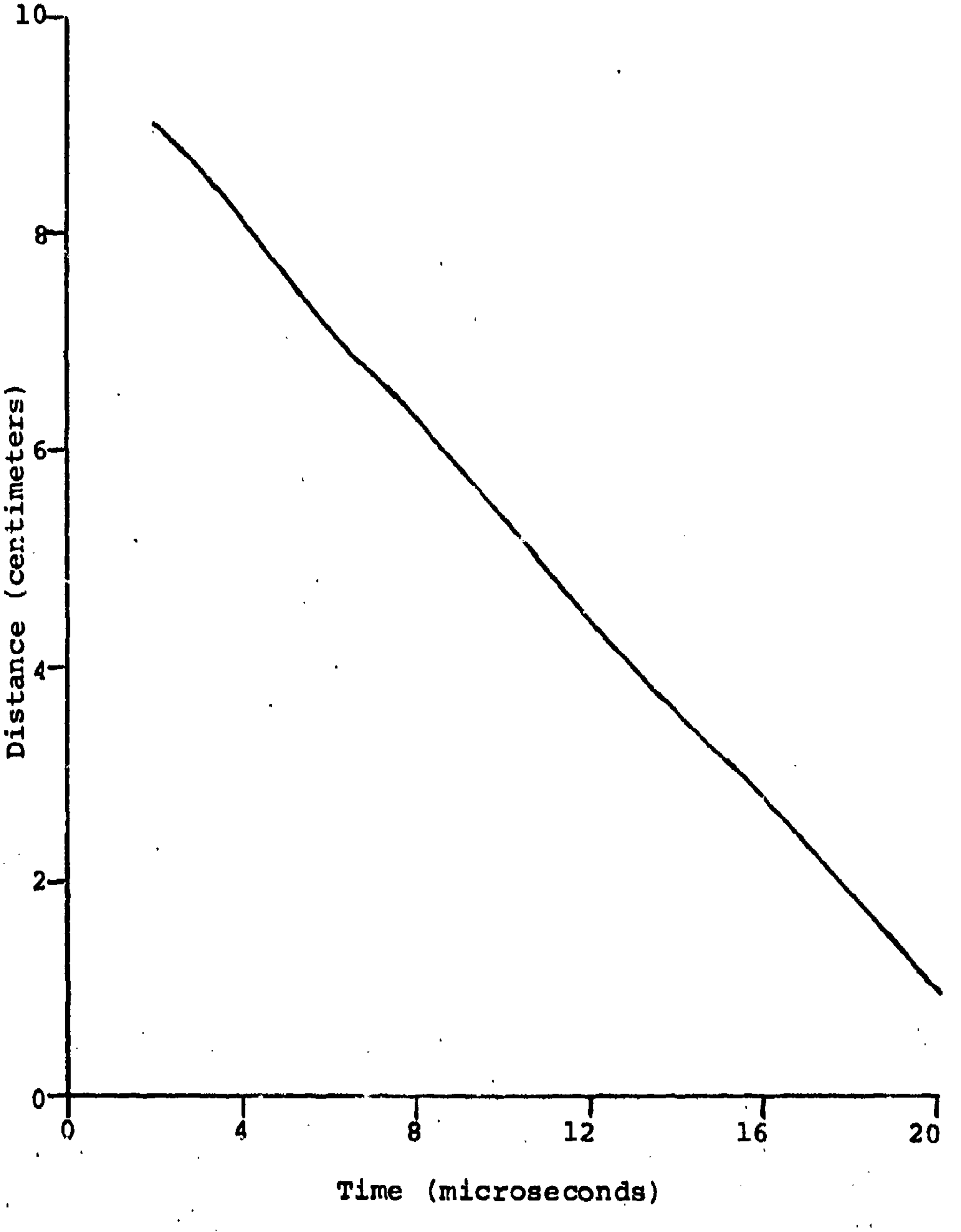


Figure 9. Shock Position vs Time for the Rectangular Compression of Uranium

Table IV

Comparison of HUFF's Compression of Uranium
with the Rankine-Hugoniot Parameters
and BEERAY's Results

Time = $1\text{x}10^{-5}$ sec (425 Time Iterations for HUFF)

	BEERAY	Standard	HUFF	%Deviation
Shock Pressure (megabars)	1±0.06	1	1.02±0.01	2
Shock Velocity (10^5cm/sec)	-4.42	-4.44	-4.44±0.37	0
Shock Density (gm/cm^3)	25.5±0.3	25.465	25.73±0.03	1
Particle Velocity (10^5cm/sec)	-1.24±0.021	-1.22	-1.26±0.01	3

(Ref 1:39)

V. Conclusions and Recommendations

HUFF is a simple, stable hydrodynamic computer code based on conservation principles. The computational time is relatively short providing results that are reasonably accurate. The conclusions drawn from the results presented in Chapter IV will be listed first followed by the recommendations.

Conclusions

The conclusions based on the results of the two sample problems are:

1. HUFF is a relatively fast computer code because the spatial mesh is limited to 100 cells, the equations of state are simple, and a variable time step control is employed to optimize the time step. In the blast problem rezoning the mesh at later times results in a significant reduction in the computation time when propagating the shock to great distances.
2. HUFF provides reasonably accurate results when propagating strong shocks. The results are within 13 percent for pressure and density, 5 percent for material velocity, and 2 percent for shock position in the blast problem and within 2 percent for all parameters (position, density, velocities) in the compression problem.
3. HUFF is a simple code due to the techniques of modular programming used in its development and the reduction of input data required to operate the sample programs.
4. HUFF is a flexible code capable of solving strong shock

propagation within larger codes or solving additional problems with minimum alteration of the existing code. This capability is a result of the task oriented independent subroutines used in HUFF.

Recommendations

The following recommendations present the author's view concerning possible extensions to HUFF that may improve its accuracy and provide an even better code.

1. Develop additional subroutines to handle the radiation transport problem for early times after a nuclear detonation and couple these subroutines with the blast expansion problem.
2. Develop additional subroutines to handle the neutronics involved in exploding the compressed uranium and couple these subroutines with the compression problem in HUFF. An additional equation of state may be necessary to handle the extremely high resulting temperatures and pressures.
3. Combine the first two recommendations by coupling the energy output from the explosion to the expansion problem providing a complete computer code that calculates the entire explosion sequence.

Bibliography

1. Anderson, Clarence L. BEERAY, a One-Dimensional Hydrodynamics Code for Strong Shocks. Master's thesis. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1975. (AD-A021 951).

2. Bridgman, Charles J. The Physics of Nuclear Explosives, Chapter 2. (Class notes). Wright-Patterson AFB, Ohio: Air Force Institute of Technology, September 1975.

3. Doan, L. R. and G. H. Nickel. A Subroutine for the Equation of State of Air. RTD (WLR) Tm-63-2. Kirtland AFB, New Mexico: Air Force Weapons Laboratory, May 1963.

4. Glasstone, Samuel, editor. The Effects of Nuclear Weapons (Third Edition). Washington: United States Atomic Energy Commission, 1977.

5. Harlow, Francis H. and Anthony A. Amsden. Fluid Dynamics. LA-4700. Los Alamos, New Mexico: Los Alamos Scientific Laboratory of the University of California, June 1971.

6. Kinslow, Ray, editor. High-Velocity Impact Phenomena. New York: Academic Press, Inc., 1970.

7. Needham, Charles E., et al. Nuclear Blast Standard (1KT). AFWL-TR-73-55 (Revised Edition). Kirtland AFB, New Mexico: Air Force Weapons Laboratory, April 1975.

8. Richtmyer, Robert D. and K. W. Morton. Difference Methods for Initial-Value Problems (Second Edition). New York: Interscience Publishers, 1967.

9. Roberts, K. V. and N. O. Weiss. "Convective Difference Schemes," Mathematics of Computation, Vol. 20, (272-299). Providence, Rhode Island: American Mathematical Society, April 1966.

10. Whitaker, William A., et al. Theoretical Calculations of the Phenomenology of HE Detonations, Vol. 1. AFWL-TR-66-141, Vol. 1. Kirtland AFB, New Mexico: Air Force Weapons Laboratory, November 1966.

11. Yourdon, E. Techniques of Program Structure and Design, Chapter 3. Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

Appendix A

The HUFF Code

This appendix describes the HUFF computer code written in the CDC FORTRAN Extended 4.4 language. The first section discusses in general the program flow and defines the specific task of each subroutine. The final section defines the primary program variables. A complete listing of program HUFF is contained in Appendix E.

Program Flow

The HUFF code contains seven subroutines and two function subprograms. Each subroutine performs a specific task in solving the assigned problem. All variables common among subroutines are passed through the appropriate argument lists.

Main Program. The main program is very short. Its basic purpose is to identify the input, output, and local files used by HUFF throughout the program. The main program calls subroutine FLAGS and then transfers control of the program to subroutine CONTROL.

Subroutine FLAGS. Subroutine FLAGS reads two input data records that define the physical operation of the program. These "flags" are used during the program to identify the program flow sequence.

Subroutine CONTROL. Subroutine CONTROL controls the program execution. All the primary task oriented programs are called in the proper sequence by subroutine CONTROL.

Subroutine BLAST. Subroutine BLAST is the "problem defining subroutine" for the blast problem. This subroutine has two entry points. A call to BLAST will set the initial conditions of the problem. A call to EXPAND will enter subroutine BLAST and calculate additional parameters for output from the program.

Subroutine SQUEEZE. Subroutine SQUEEZE is the "problem defining subroutine" for the compression problem. This subroutine also has two entry points. A call to SQUEEZE sets the initial conditions while a call to CRUSH enters later in the subroutine and calculates the same parameters computed by EXPAND in the blast problem.

Subroutine HYDRO. Subroutine HYDRO is the heart of HUFF. All the hydrodynamic shock propagation calculations are accomplished by this subroutine. The subroutine begins by calculating a proper time step based on the current material properties. HYDRO then calculates the new material properties advancing the mesh through one time step. The subroutine contains an additional time step control to optimize the computed time step. This additional "hydromatic" control varies the computed time step forcing the primary shock to cross a cell boundary each 5-10 cycles through subroutine HYDRO.

Subroutine REZONE. Subroutine REZONE extends the distance the expanding blast shock can be followed by rezoning the mesh each time the shock reaches cell 94. This subroutine carefully conserves the material instabilities present before rezone so they match those present after rezone. This enables the rezoned shock to continue its propagation with minimum interruption.

Function EIA. Function EIA determines the proper internal energy corresponding to a given pressure and density. This subroutine is necessary since gamma is a function of internal energy and internal energy is not directly computed and maintained by subroutine HYDRO. Subroutine

HYDRO and subroutine REZONE both use function EIA.

Function VGAMMA. Function VGAMMA contains the HUFF rough fit for the variable gamma function used in the equation of state for air. This function determines gamma based on a given internal energy and density. Function VGAMMA is used in both subroutines HYDRO and REZONE.

Subroutine COPY. Subroutine COPY handles the recording of results for the program. A complete listing of all the program arrays are printed on a local file called files for each result time requested by the user. A rough plot of the requested results is also made with appropriate calls to PLOTSM.

Subroutine PLOTSM. The last subroutine contained in HUFF is subroutine PLOTSM. This subroutine provides a rough plot of the program arrays on the output file. A rough plot of all the requested results for aid in analysis of the output is provided.

Obtaining Formal Output. When the program execution is complete, the requested results are contained on two local files. A file named files contains the requested information. A file named Review contains summary information. The information can be read from these files and minipulated into any desired form. An additional program called Results is listed in Appendix E that will read the data contained on the local files and request the appropriate formal plots.

Main Program Variables

This section contains a listing of all the program variables used in HUFF. Many of the acronyms suggest their specific use in the program. Unless otherwise indicated, all program variables are in the gram-centimeter-second system of units. The program variables are:

A:	Temporary storage variable.
C:	Adiabatic sound speed in a solid.
CA:	The square of the number of cells over which the shock discontinuity is spread (a^2 in Eq (19)).
CK(1)/CYCLE/ICYCLE:	Counters for the number of cycles through subroutine HYDRO.
CK(2)/TIME:	Keeps track of total real time (sec) during the solution.
CK(3)/DT:	The time step value (sec).
CK(4)/POS:	Shock position.
CK(5)/OP:	Peak shock overpressure.
CK(6)/OD:	Peak shock overdensity.
CK(7)/VMAX:	Peak material velocity at the shock.
D(I):	Cell centered density array.
DMAX:	Maximum cell centered density at the shock.
DTJ:	Temporary local time step storage variable.
DX:	Original mesh spacing of the array.
EI/EIE:	Internal energy in ergs/gram and 10^{10} ergs/gram respectively.
EKJ1/EKJ2/EKI1:	Kenetic energy storage variables.
EL:	Natural logrithm of EIE.
ETJ1/ETJ2/ETI1:	Total energy storage variables.
FLAG(2)/IF2:	Flag for the number of results times requested.
FLAG(3)/IF3:	Flag for the maximum number of allowed cycles through subroutine HYDRO.
FLAG(4):	Flag used to identify the type of problem.
GAMMA:	Storage variable for the Gruneisen constant or gamma.
GM1/GMONE:	Storage variable for gamma minus one.

HIOLD/HINEW: Temporary storage variables used for I_H from the solid equation of state.

HPOLD/HPNEW: Temporary storage variables used for P_H from the solid equation of state.

IALPHA/L: Problem geometry factors.

ICOUNT/KFIFTY: Loop counters.

IP/N1/NSAVE: Shock grid position.

MN/M: Used as a variable length for do loops in subroutine HYDRO.

MP1: M plus one.

NM5/NP5: N1 minus 5 and plus 5 respectively.

OPOS: Old position of the shock.

P(I): Material property array for cell centered pressures.

P2/PX: Temporary pressure storage variables.

PRESS: Input pressure for the compression problem.

PUD(I,J): Temporary storage array for cell centered pressure, velocity, and density respectively.

PMAX: Maximum pressure at the shock front.

PNORM: Normal atmospheric pressure at the burst altitude.

PBLAST: Input pressure for the blast problem.

Q(I): Material property array for the artificial cell centered artificial viscosity.

QMAX: Maximum viscosity at the shock front. This occurs at the location of the shock by definition.

RHOZ: Ratio of D(I)/RHO.

RHO: Normal material density.

RADIUS: Initial radius of the solid in the compression problem.

S: Temporary storage variables for various functions of gamma.

SP/SD/ST: Altitude scaling parameters for pressure, density, and time respectively.

TSTOP: Real time limit for the problem.

TFRAC: A variable fraction used to place additional control on the size of the time step to increase program efficiency.

TIMIT: Used as a loop timer in calculating shock speeds.

U(I): Material property array for the cell boundary velocities.

UJ1/UJ2/UI1: Temporary storage variables for cell centered velocity in subroutine REZONE.

VG(I): Variable gamma storage array in subroutine HYDRO.

VEL(I): Additional array used to temporarily store velocity in subroutine HYDRO.

VRHO: The normal specific density of the material.

VOLD/VNEW: Temporary storage variables for cell centered specific densities used in subroutine HYDRO.

VDOT: Specific density change in one time step.

VSVOLD/VSVNEW: Temporary storage variable used with the solid equation of state.

X(I): Material property array for the cell boundary positions.

XI: Real variable equal to the integer I.

XMJ1/XMJ2/XMI1: Temporary storage arrays for the mass of the two old cells and the new cell respectively used in subroutine REZONE.

XMOMJ1/XMOMJ2/XMOMI1: Temporary storage arrays for the momentum of the two old cells and the new cell respectively used in subroutine REZONE.

XNORM: A variable used in a scaling function representing the normal value of the property to be scaled.

XD: A variable used in a scaling function representing the altitude scaling factor.

YKT: Input weapon yield in kilotons in the blast problem.

Appendix B

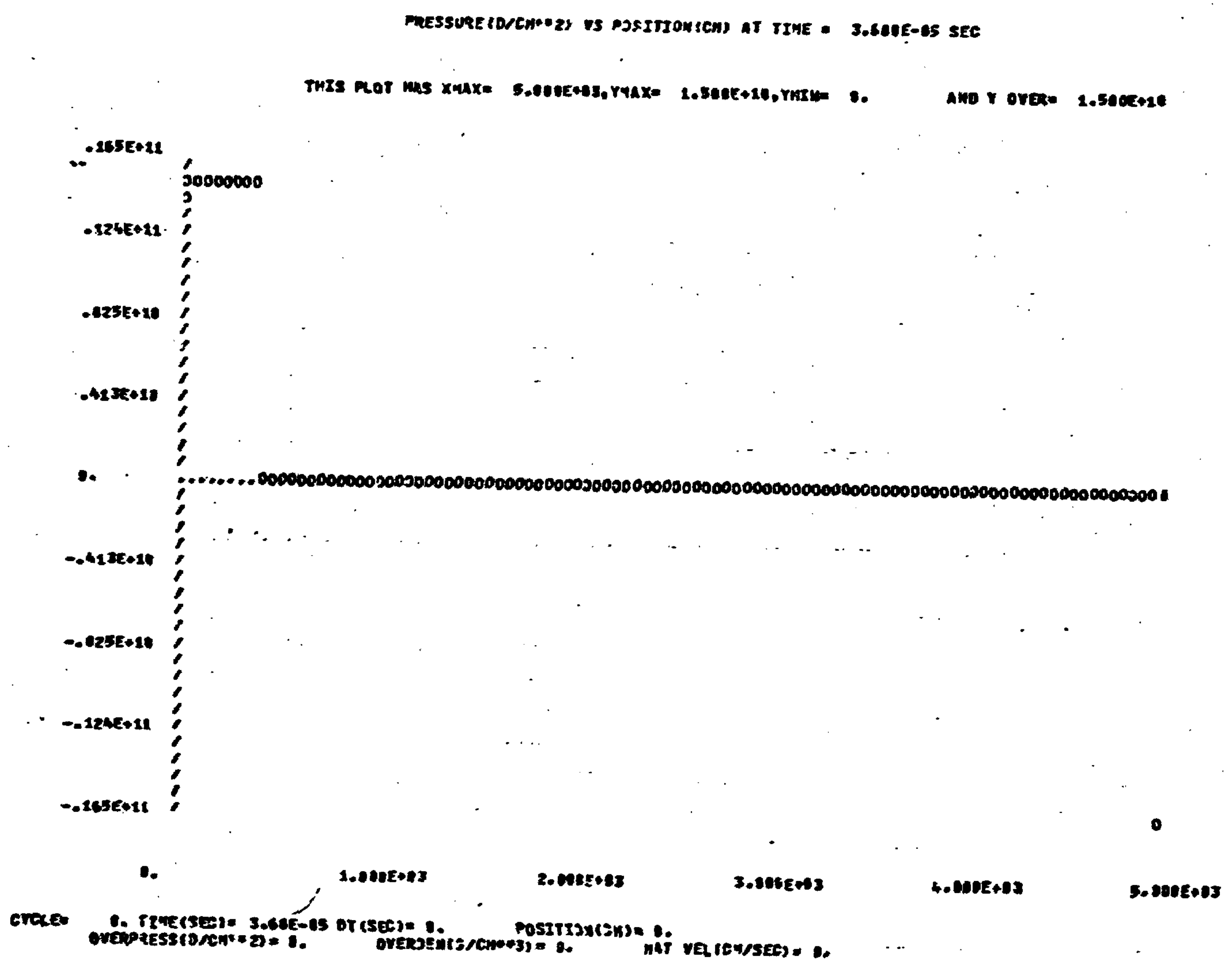
Sample Problem, 1KT Explosion

The data for HUFF is always read by a list directed format to simplify the input cards.

The input cards for this problem are as follows where the items in parentheses are not part of the data deck:

```
10 3000 1 5.0                                        (Card 1)
.003 .006 .01 .02 .05 .1 .2 .5 1. 2.                 (Card 2)
1. 1. 1. 1.                                          (Card 3)
```

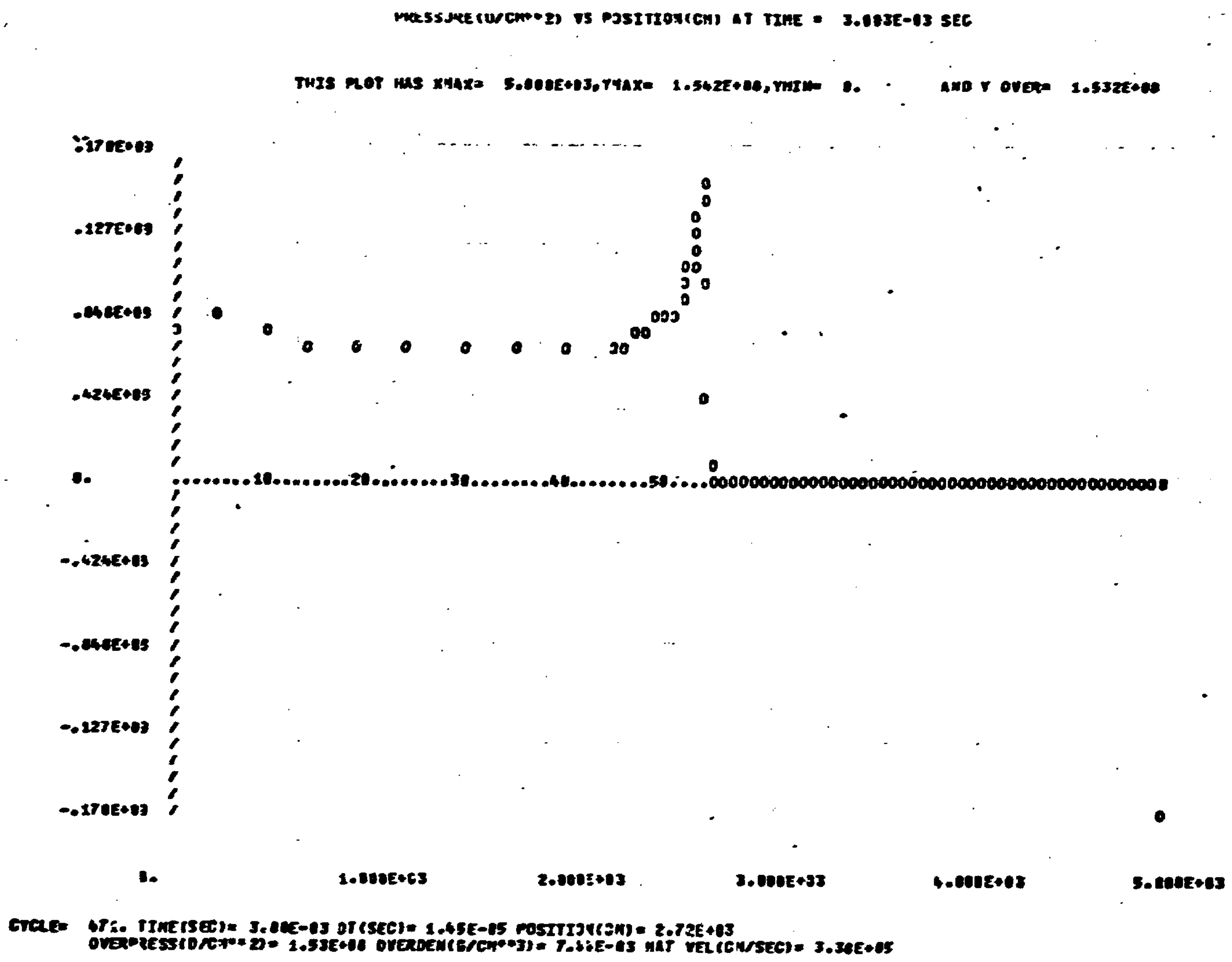
Figures 4 through 6 in Chapter IV of the text and Figures 10 through 18 of this appendix represent sample output for this problem.



```
PRESSURE(D/CM**2) VS POSITION(CM) AT TIME =  3.600E-05 SEC

THIS PLOT HAS XMAX=  5.000E+03,YMAX=  1.500E+10,YMIN=  0.        AND Y OVER=  1.500E+10
```

```
CYCLE=    0. TIME(SEC)=  3.60E-05 DT(SEC)= 0.        POSITION(CM)= 0.
     OVERPRESS(D/CM**2)= 0.        OVERDEN(G/CM**3)= 0.        MAT VEL(CM/SEC)= 0.
```

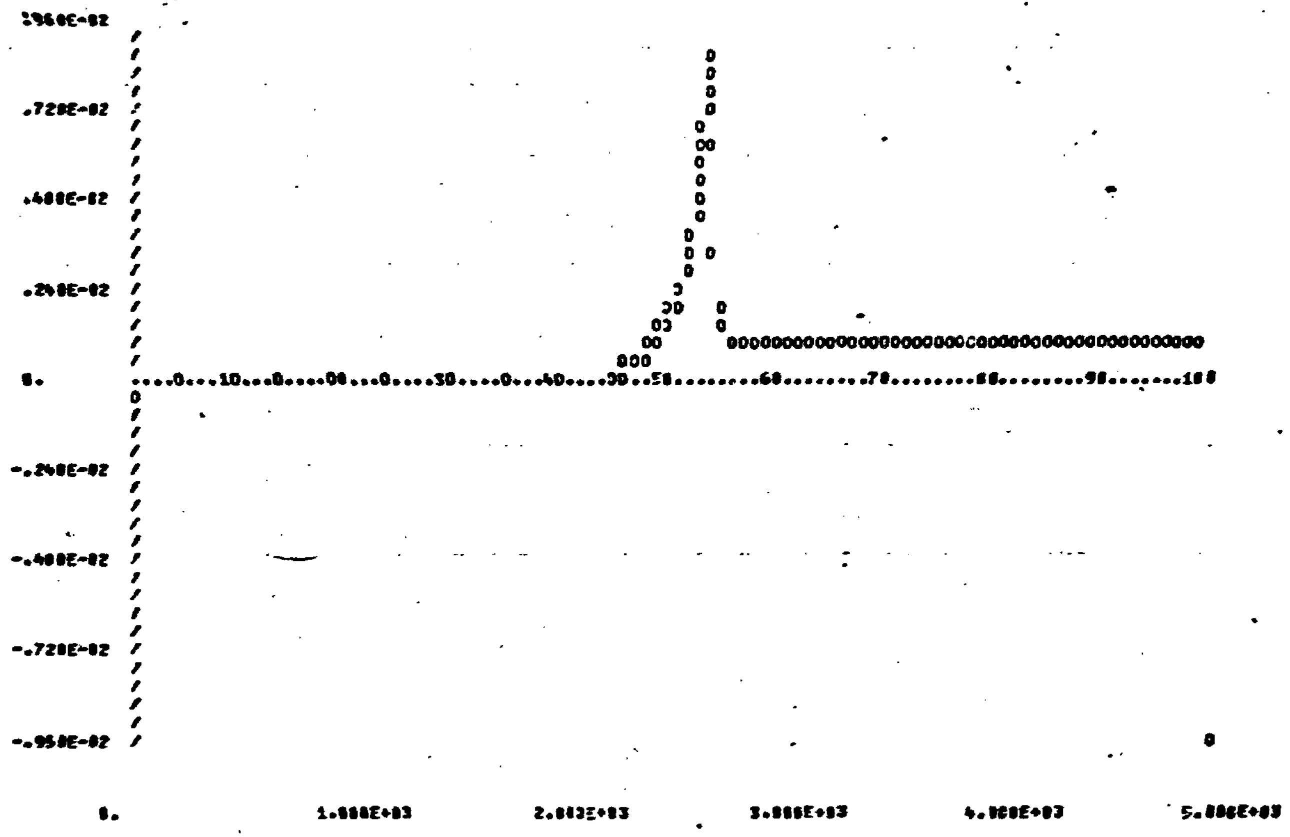
Figure 10. Initial Pressure Condition for 1KT Burst



```
PRESSJRE(U/CM**2) VS POSITION(CM) AT TIME =  3.883E-03 SEC

THIS PLOT HAS XMAX=  5.000E+03,YMAX=  1.542E+08,YMIN=  0.        AND Y OVER=  1.532E+08

CYCLE=  471. TIME(SEC)= 3.80E-03 DT(SEC)= 1.45E-05 POSITIJN(CM)= 2.72E+03
        OVERPRESS(D/CM**2)= 1.53E+08 OVERDEN(G/CM**3)= 7.44E-03 MAT VEL(CM/SEC)= 3.30E+05
```

Figure 11. Pressure for 1KT Burst ($3x10^{-3}$sec)

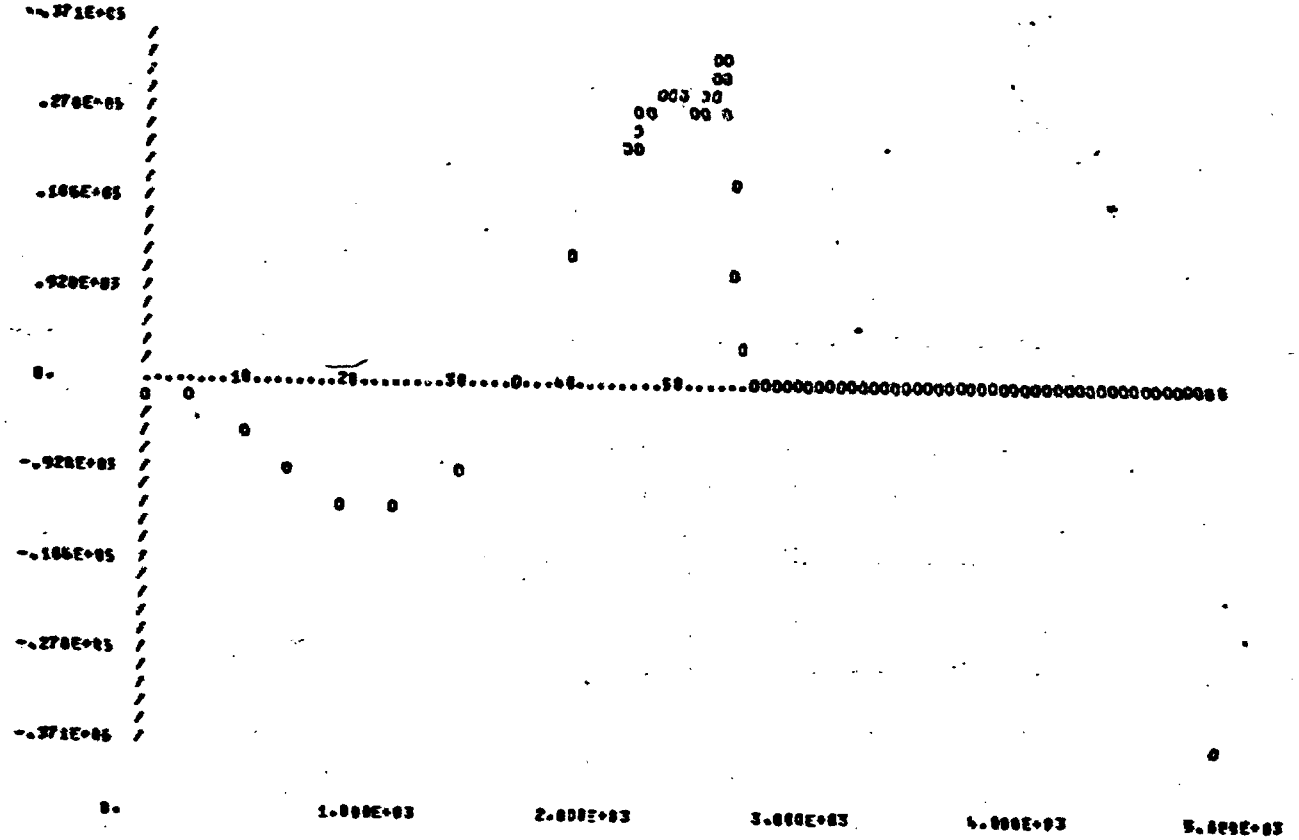
```
DENSITY(G/CM**3) VS POSITION(CM) AT TIME =  3.003E-03 SEC

THIS PLOT HAS XMAX=  5.000E+03,YMAX=  9.731E-03,YMIN=  0.        AND Y OVER=  7.430E-03
```



```
CYCLE=  471. TIME(SEC)= 3.00E-03 DT(SEC)= 1.45E-05 POSITION(CM)= 2.72E+03
        OVERPRESS(D/CM**2)= 1.53E+06 OVERDEN(G/CM**3)= 7.43E-03 MAT VEL(CM/SEC)= 3.30E+05
```

Figure 12. Density for 1KT Burst ($3x10^{-3}$sec)

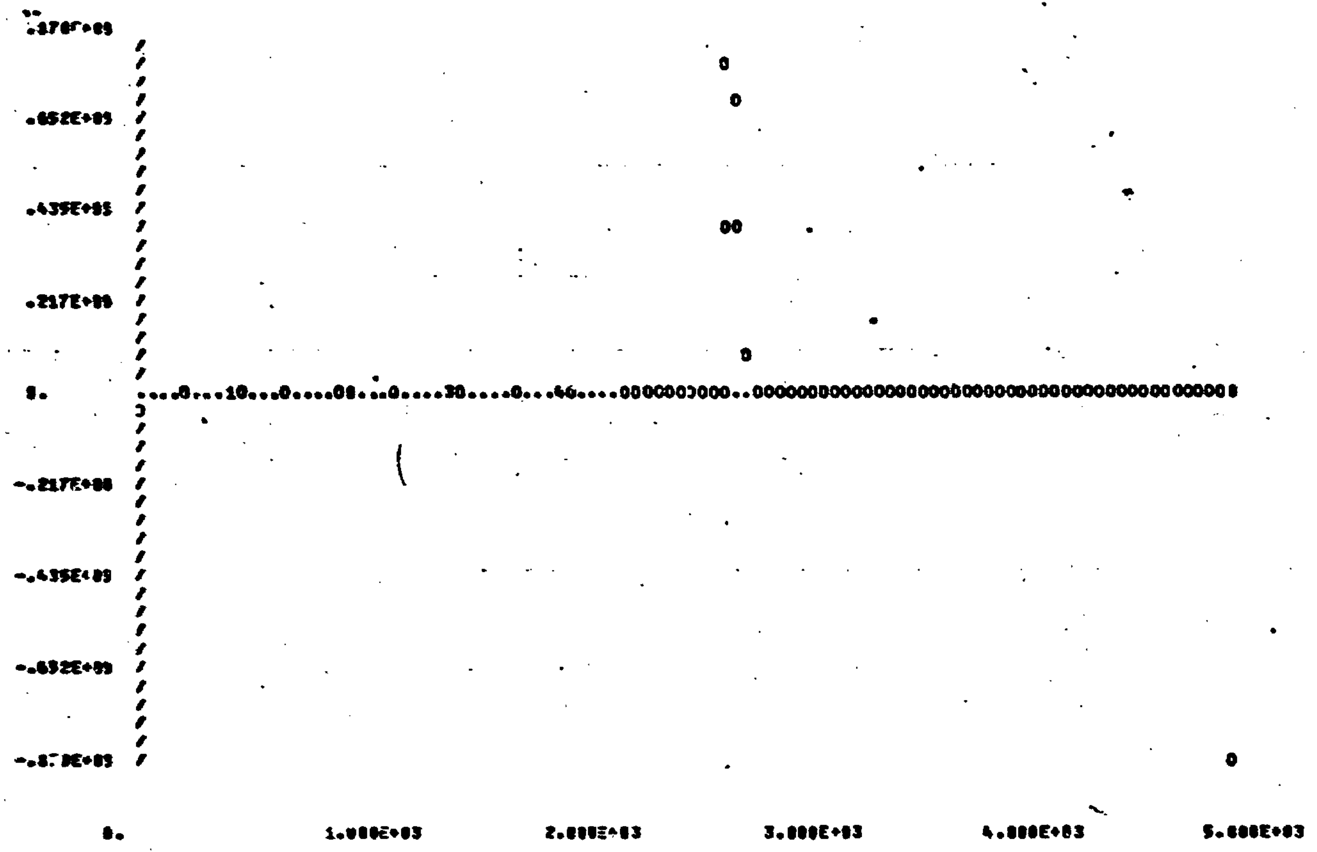
```
VELOCITY(CM/SEC) VS POSITION(CM) AT TIME =  3.003E-03 SEC

THIS PLOT HAS XMAX=  5.000E+03,YMAX=  3.376E+05,YMIN= -1.246E+05 AND Y OVER=  3.376E+05
```



```
CYCLE=  471. TIME(SEC)= 3.00E-03 DT(SEC)= 1.45E-05 POSITION(CM)= 2.72E+03
        OVERPRESS(D/CM**2)= 1.53E+06 OVERDEN(G/CM**3)= 7.44E-03 MAT VEL(CM/SEC)= 3.38E+05
```

Figure 13. Material Velocity for 1KT Burst ($3x10^{-3}$sec)

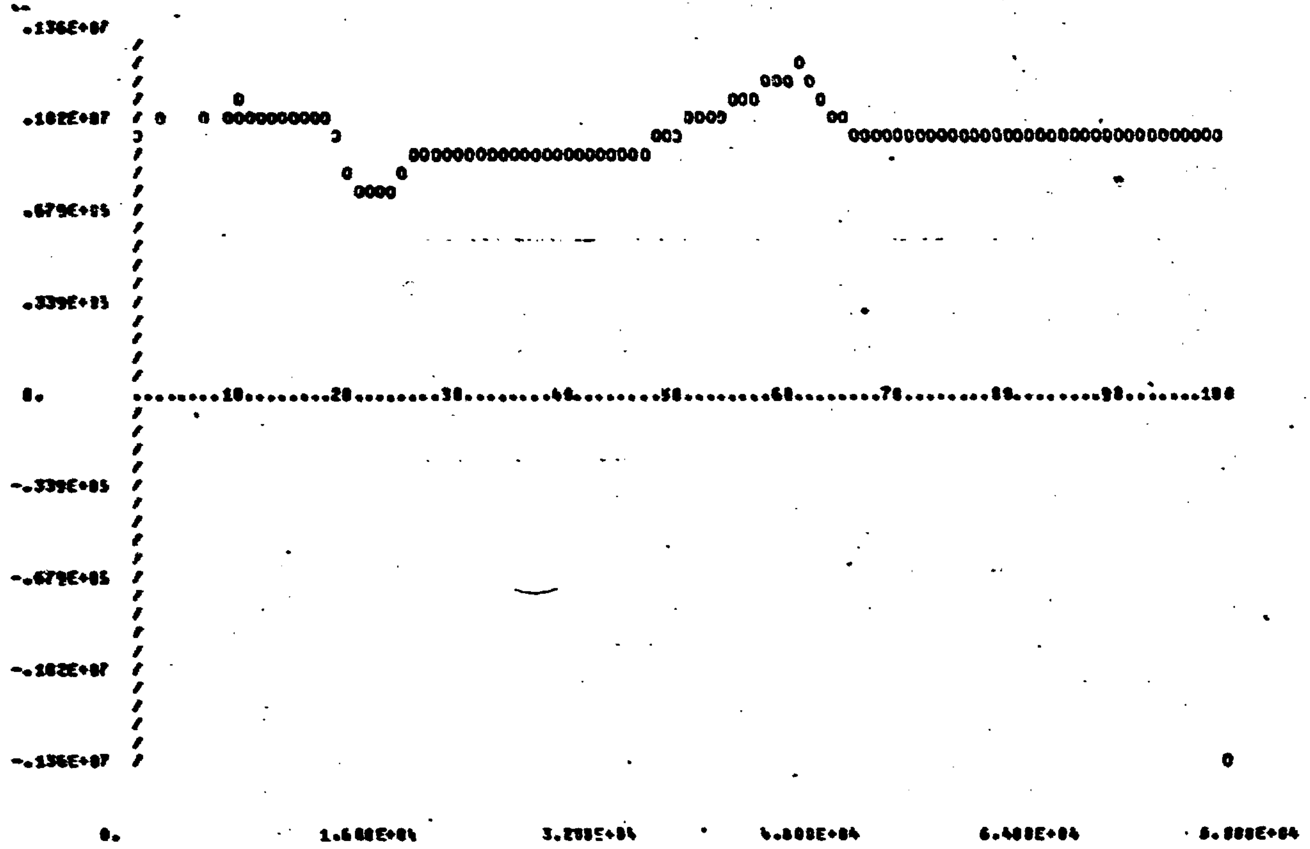
```
VISCOSITY(D/CM**2) VS POSITION(CM) AT TIME =  3.003E-03 SEC

THIS PLOT HAS XMAX=  5.000E+03,YMAX=  7.900E+07,YMIN=  0.        AND Y OVER=  7.900E+07
```



```
CYCLE=  471. TIME(SEC)= 3.00E-03 DT(SEC)= 1.45E-05 POSITION(CM)= 2.72E+03
        OVERPRESS(G/CM **2)= 1.53E+00 OVERDEN(G/CM**3)= 7.44E-03 MAT VEL(CM/SEC)= 3.30E+05
```

Figure 14. Artificial Viscosity for 1KT Burst (3x10^{-3}sec)

```
PRESSURE(D/CM**2) VS P3SITI3N(CM) AT TIME =  1.002E+00 SEC

THIS PLOT HAS XMAX=  8.000E+04,YMAX=  1.234E+06,YMIN=  0.        AND Y OVER=  2.200E+05
```

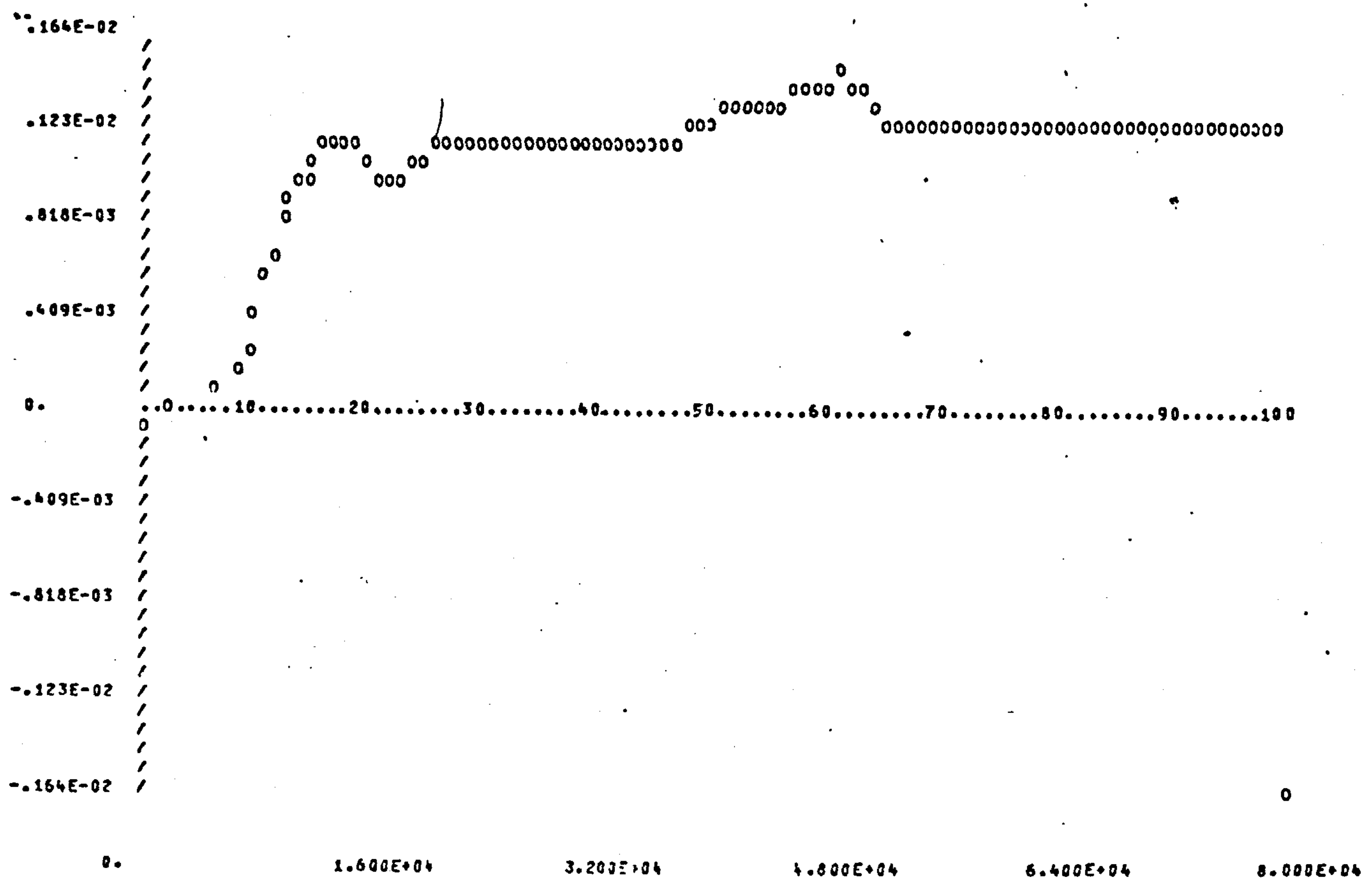


```
CYCLE= 2340. TIME(SEC)= 1.00E+00 DT(SEC)= 3.06E-03 POSITI3N(CM)= 3.05E+04
       OVERPRESS(D/CM**2)= 2.20E+05 OVERDEN(G/CM**3)= 1.34E-04 MAT VEL(CM/SEC)= 5.04E+03
```

Figure 15. Pressure for 1KT Burst (1sec)

DENSITY(G/CM**3) VS POSITION(CM) AT TIME = 1.002E+00 SEC

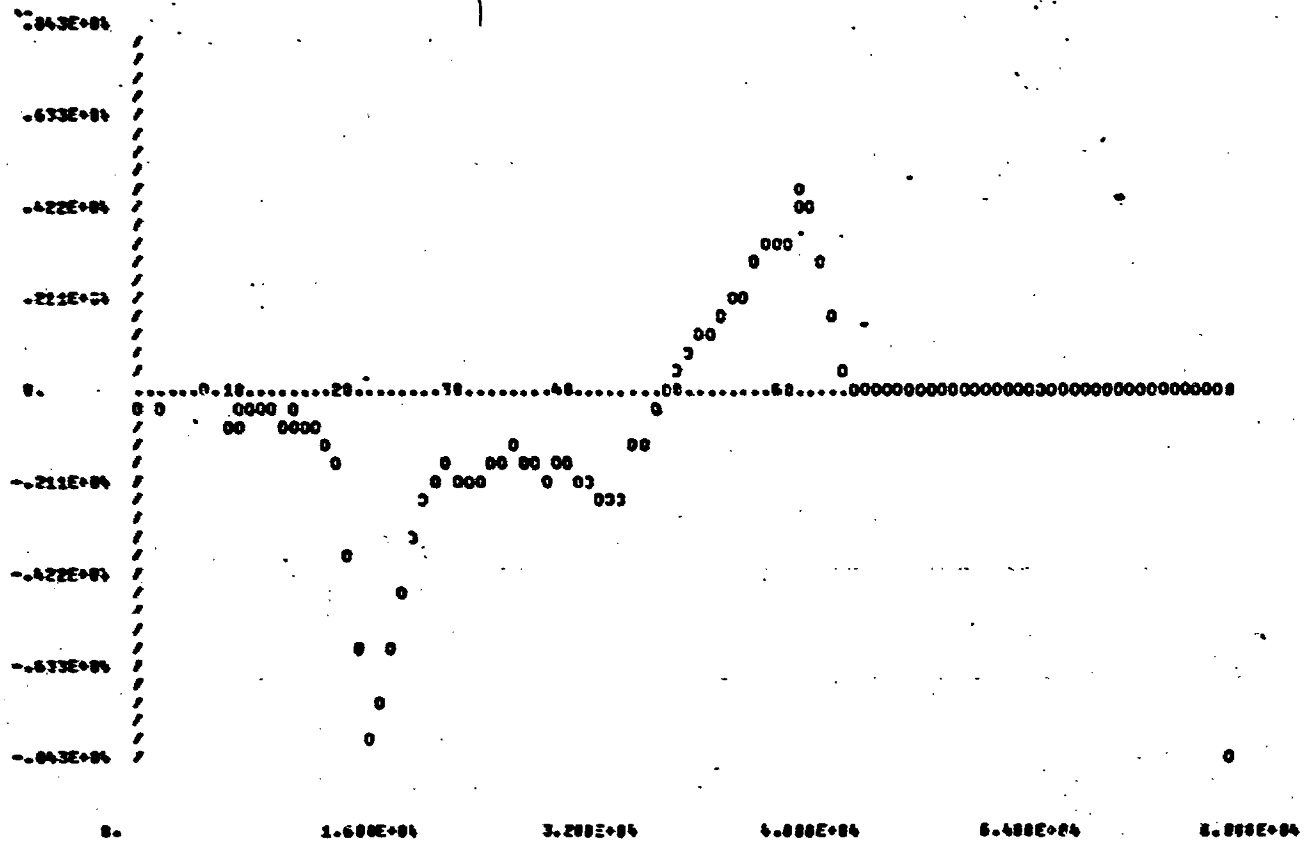
THIS PLOT HAS XMAX= 8.000E+04,YMAX= 1.487E-03,YMIN= 0.   AND Y OVER= 1.945E-04



```
CYCLE= 2340. TIME(SEC)= 1.00E+00 DT(SEC)= 3.05E-03 POSITION(CM)= 5.05E+04
       OVERPRESS(D/CM**2)= 2.20E+05 OVERDEN(G/CM**3)= 1.34E-04 MAT VEL(CM/SEC)= 5.04E+03
```

Figure 16. Density for 1KT Burst (1sec)

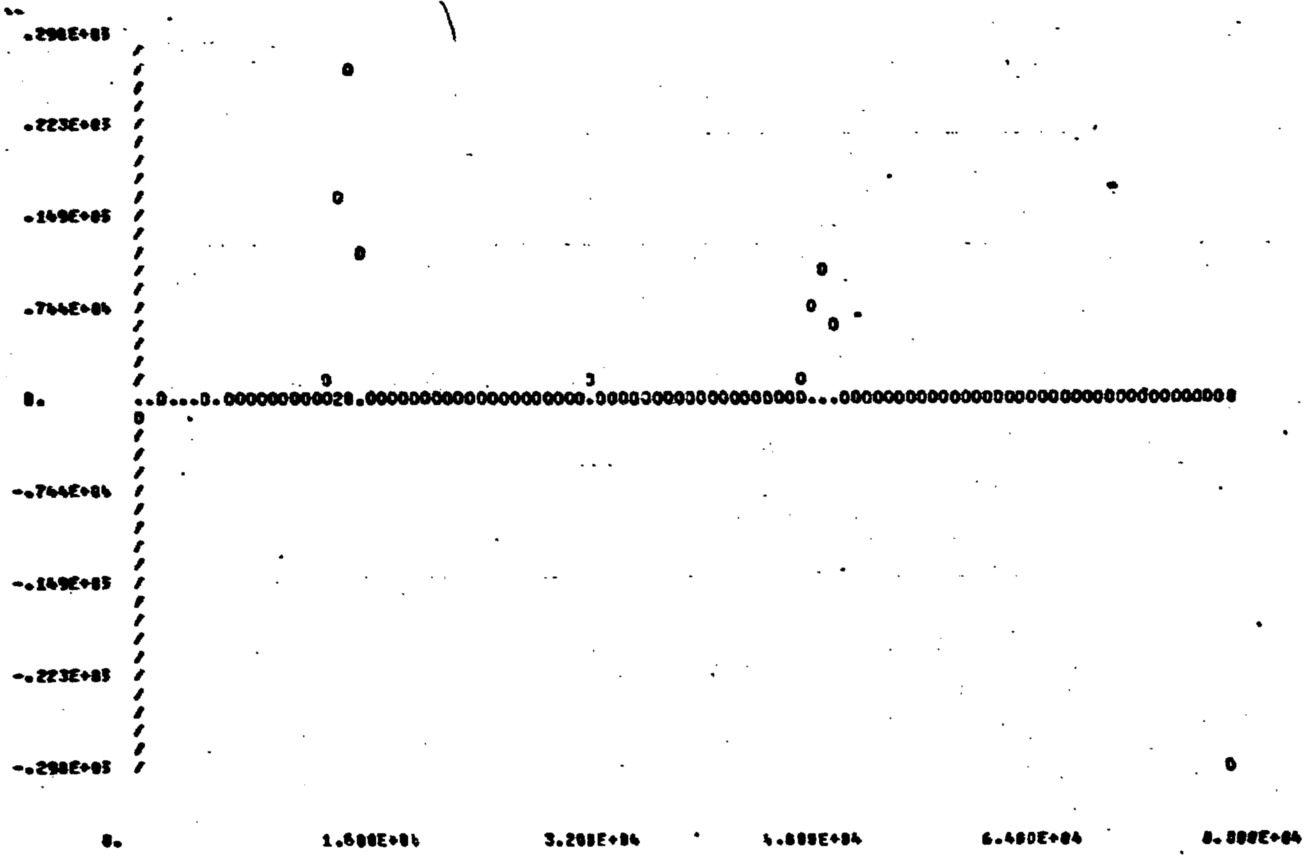
```
VELOCITY(CM/SEC) VS POSITION(CM) AT TIME =  1.002E+00 SEC

THIS PLOT HAS XMAX=  8.000E+04,YMAX=  5.839E+03,YMIN= -7.668E+03 AND Y OVER=  5.839E+03
```



```
CYCLE= 2340. TIME(SEC)= 1.00E+00 DT(SEC)= 3.06E-03 POSITION(CM)= 3.05E+04
        OVERPRESS(D/CM**2)= 2.20E+05 OVERDEN(G/CM**3)= 1.34E-04 MAT VEL(CM/SEC)= 5.04E+03
```

Figure 17. Material Velocity for 1KT Burst (1sec)

```
VISCOSITY(D/CM**2) VS POSITION(CM) AT TIME =  1.002E+00 SEC

THIS PLOT HAS XMAX=  8.000E+04,YMAX=  2.786E+04,YMIN=  0.        AND Y OVER=  2.786E+04
```



```
CYCLE= 2348. TIME(SEC)= 1.00E+00 DT(SEC)= 3.05E-03 POSITION(CM)= 5.05E+04
       OVERPRESS(D/CM**2)= 2.20E+05 OVERDEN(G/CM**3)= 1.34E-06 MAT VEL(CM/SEC)= 5.04E+03
```

Figure 18. Artificial Viscosity for 1KT Burst (1sec)

Appendix C

Sample Problem, One Megabar Compression of Uranium

The input cards for this problem in rectangular geometry are as follows where the items in parentheses are not part of the data deck:

```
10 4000 2 1.0                                         (Card 1)
1.E-5 2.E-5 3.E-5 4.E-5 5.E-5 6.E-5 7.E-5 8.E-5
      9.E-5 1.E-4                                     (Card 2)
1 2.03 18.45 2.56E5 10. 1.E12                         (Card 3)
```

The data is in list directed format to simplify defining the problem.

For a spherical compression problem change Card 3 to the following:

```
3 2.02 18.45 2.56E5 10. 1.E10                         (Card 3)
```

The compression sample problems maintain a constant applied pressure in the outer cell of the mesh throughout the problem solution. This can be changed to an impulse pressure by removing the card that redefines the outer cell pressure immediately after ENTRY CRUSH in subroutine SQUEEZE. A comment card is included in the program to identify the location of this card.

Figures 7 through 9 in Chapter IV of the text and Figures 19 through 23 of this appendix represent sample output for the rectangular geometry problem. Figures 24 through 32 of this appendix represent sample output for the case of spherical geometry.

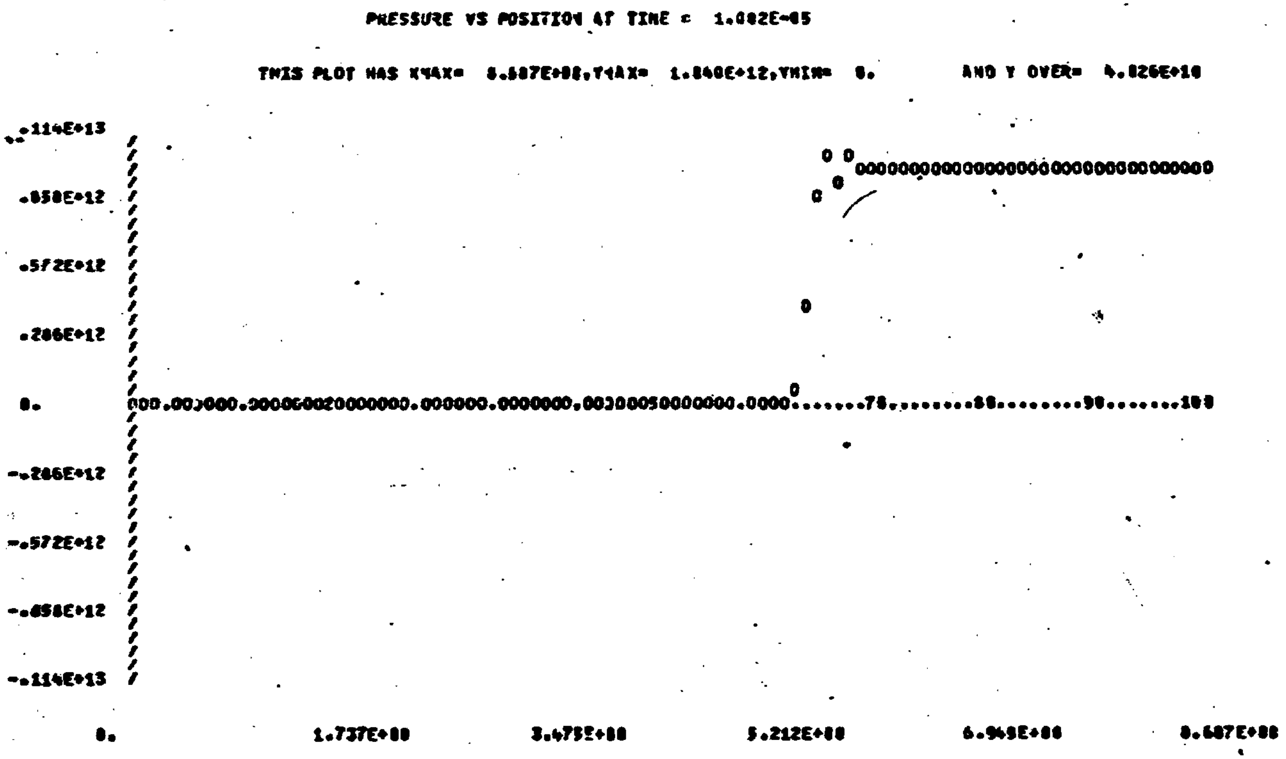


Figure 19. Pressure for Rectangular Compression of Uranium ($1x10^{-5}$sec)

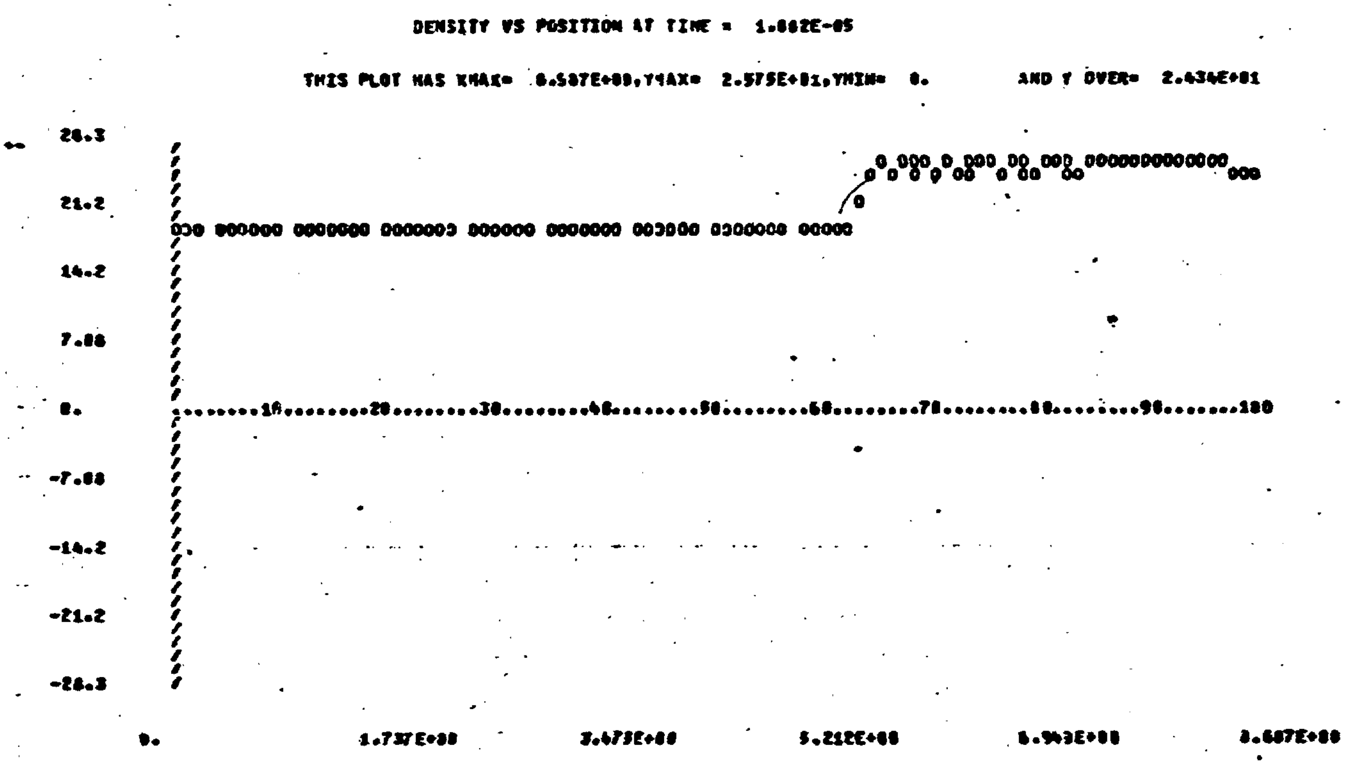


Figure 20. Density for Rectangular Compression of Uranium (1x10^{-5}sec)

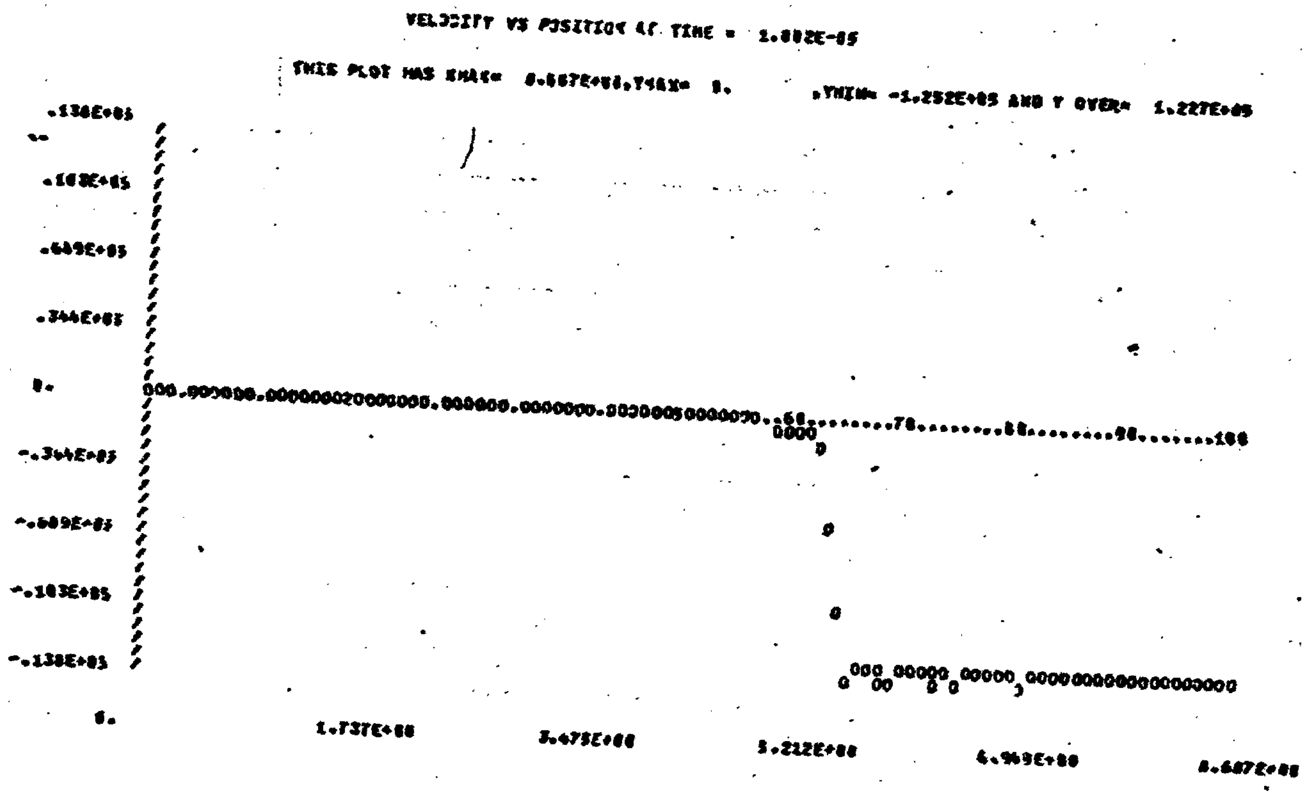


Figure 21. Material Velocity Rectangular Compression of Uranium ($1x10^{-5}$sec)

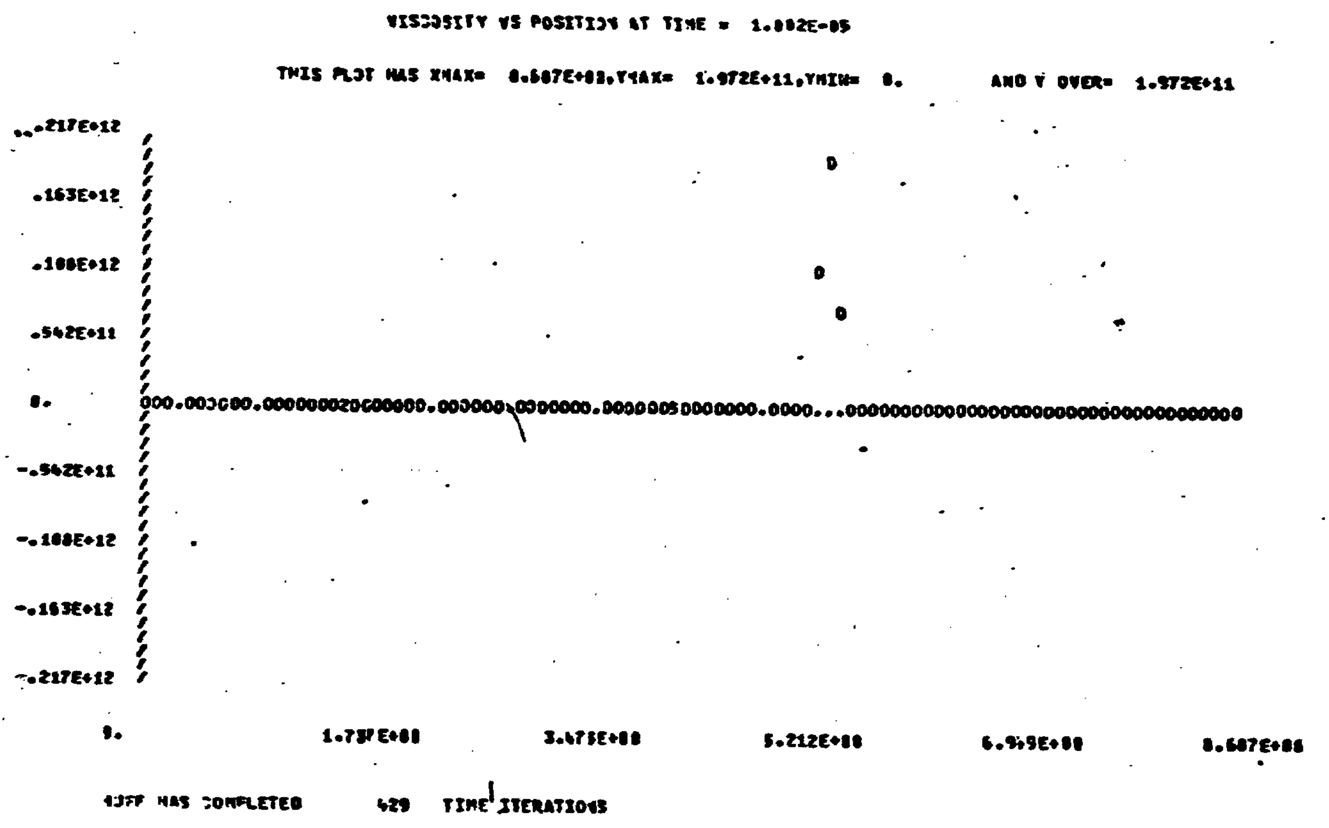


Figure 22. Artificial Viscosity for Rectangular Compression of Uranium (1x10^{-5}sec)

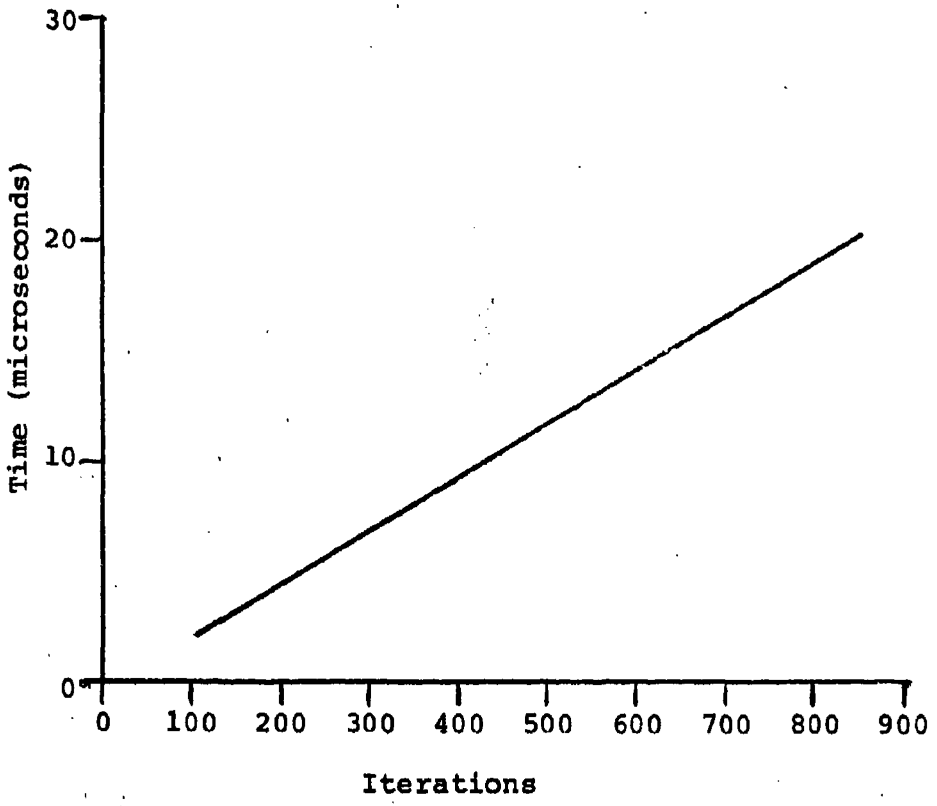
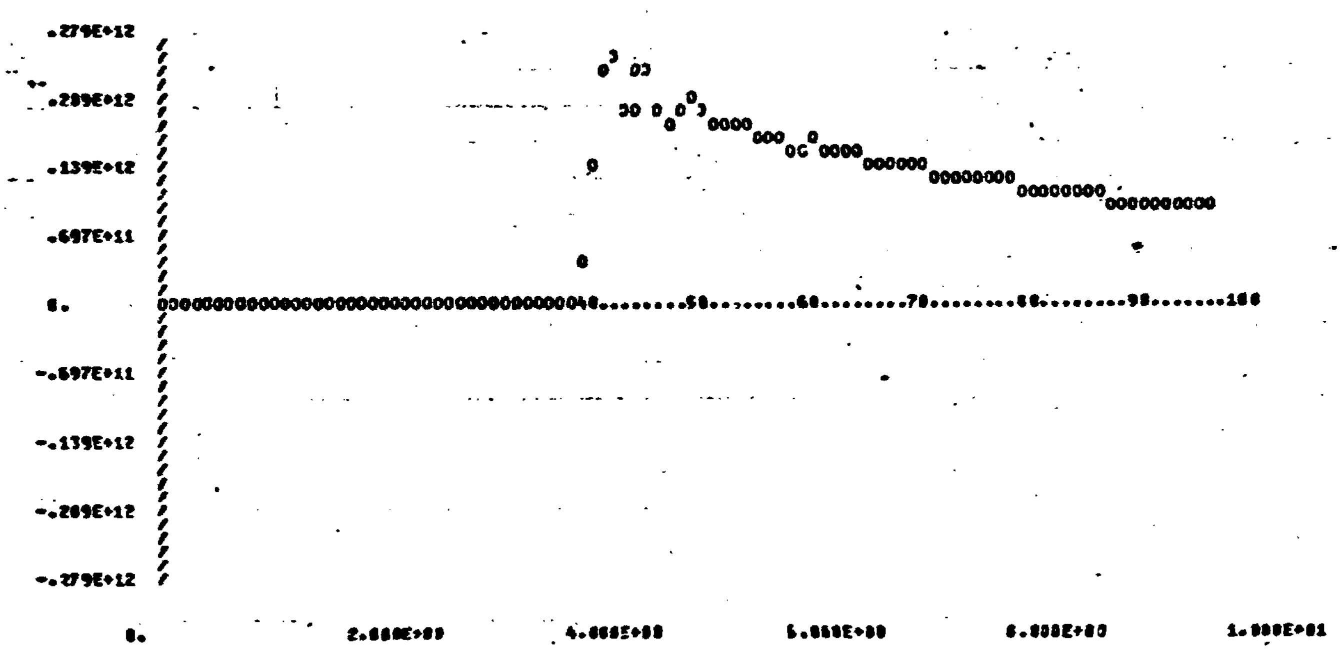


Figure 23. Time vs Iterations for Rectangular Compression of Uranium



```
PRESSURE VS POSITION AT TIME =  2.002E-05

THIS PLOT HAS XMAX=  1.000E+01,YMAX=  2.534E+11,YMIN=  0.        AND Y OVER=  1.534E+11
```

Figure 24. Pressure for Spherical Compression of Uranium ($2x10^{-5}$sec)

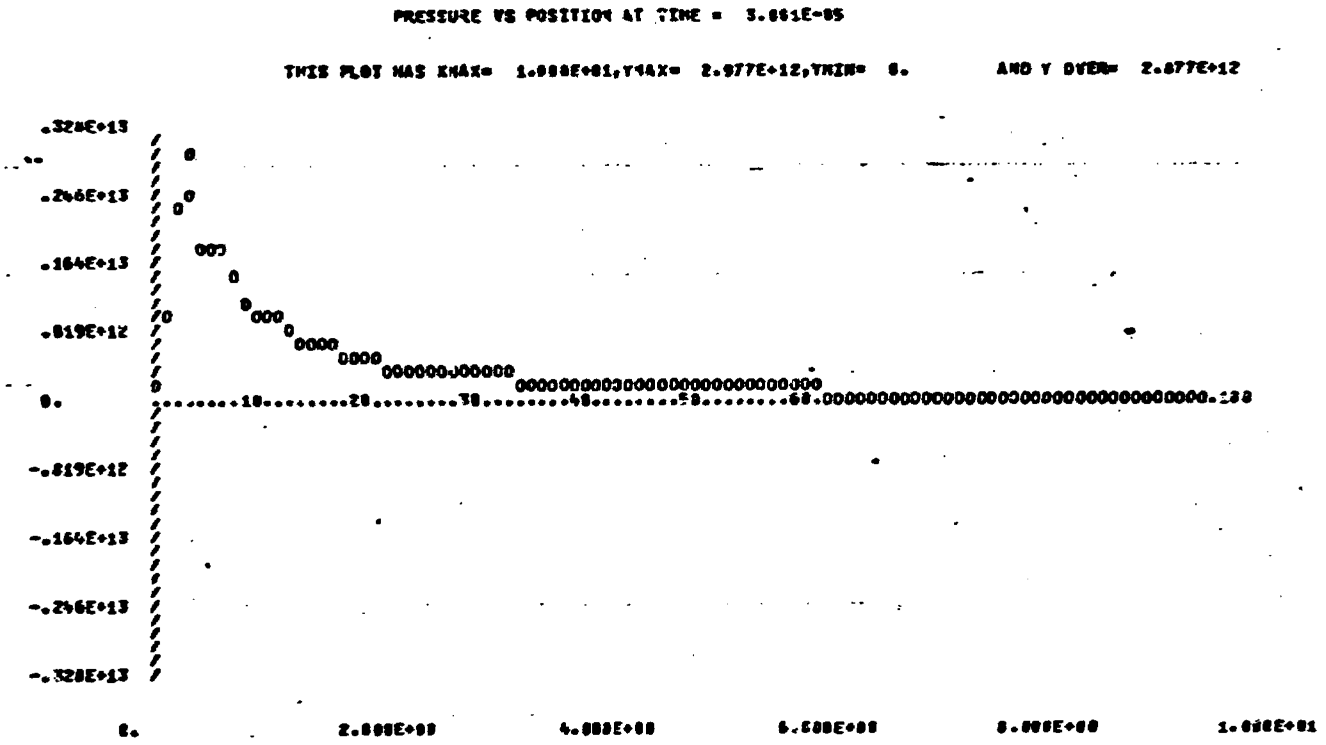


Figure 25. Pressure for Spherical Compression of Uranium (3×10^{-5}sec)

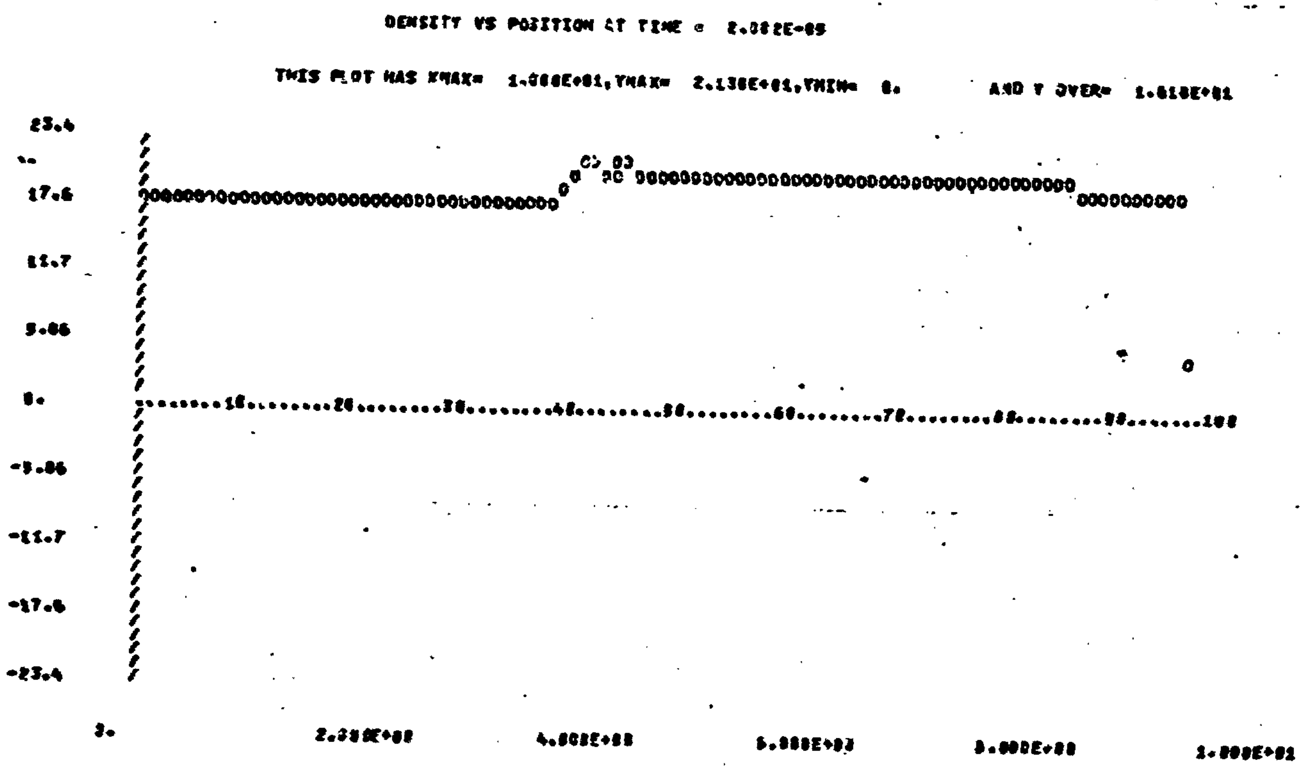


Figure 26. Density for Spherical Compression of Uranium (2x10^{-5}sec)

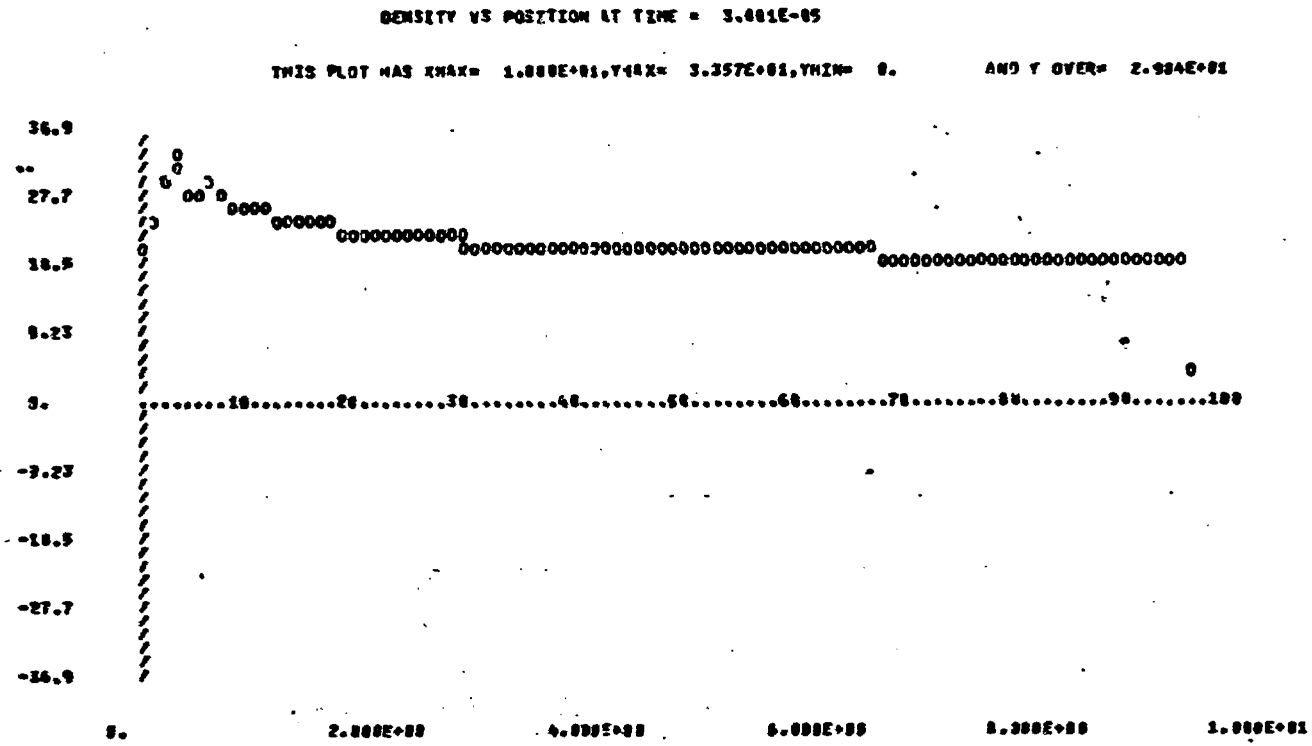


Figure 27. Density for Spherical Compression of Uranium ($3x10^{-5}$sec)

```
VELOCITY VS POSITION AT TIME =  2.000E-05

THIS PLOT HAS XMAX=  1.000E+01,YMAX=  0.       ,YMIN= -4.272E+04 AND Y OVER=  7.961E+03
```

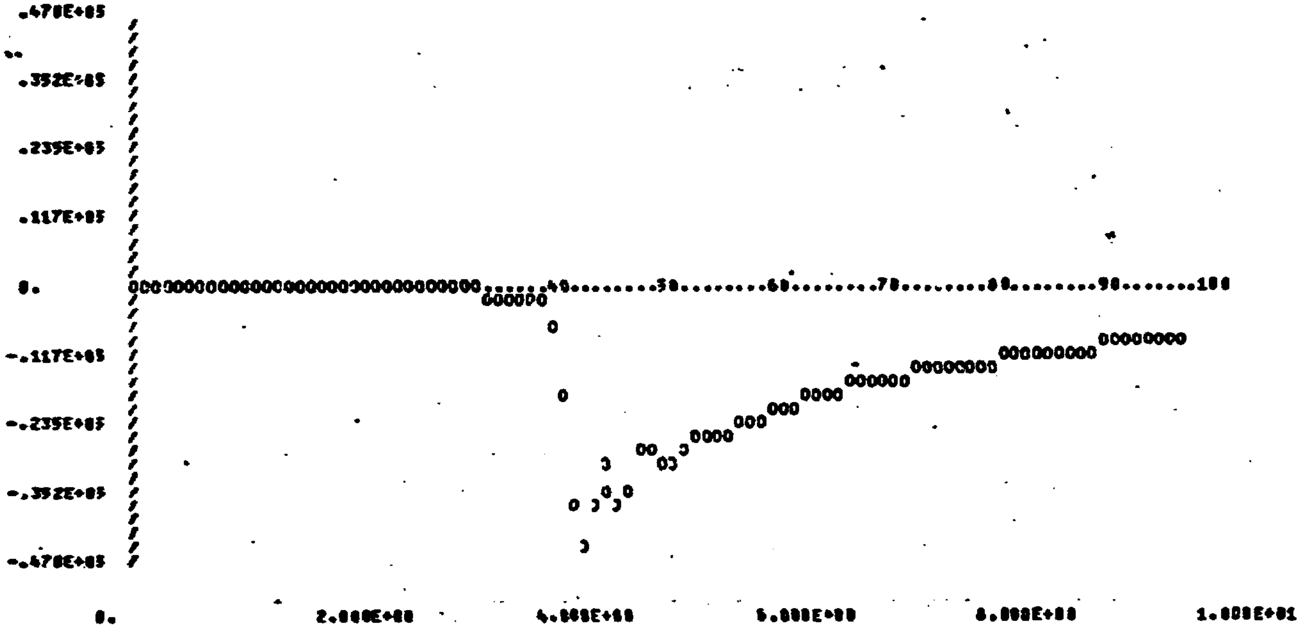
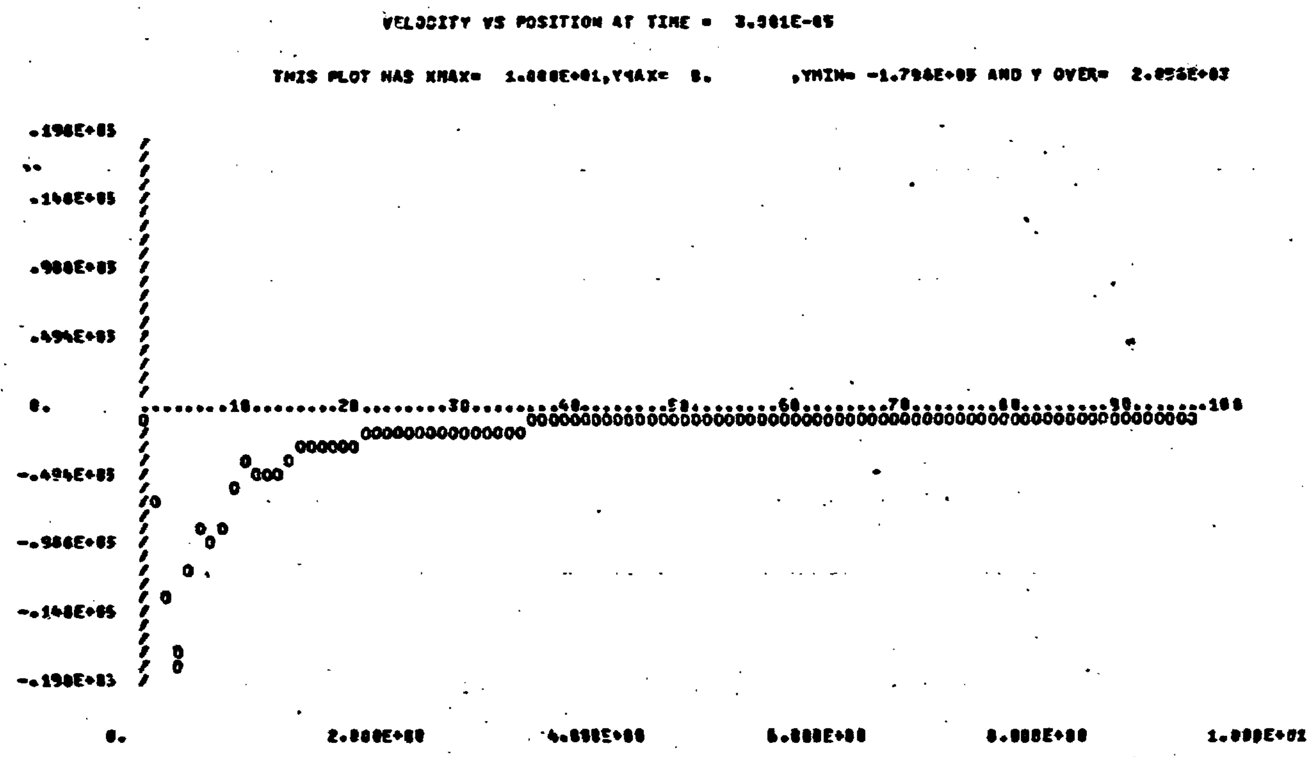
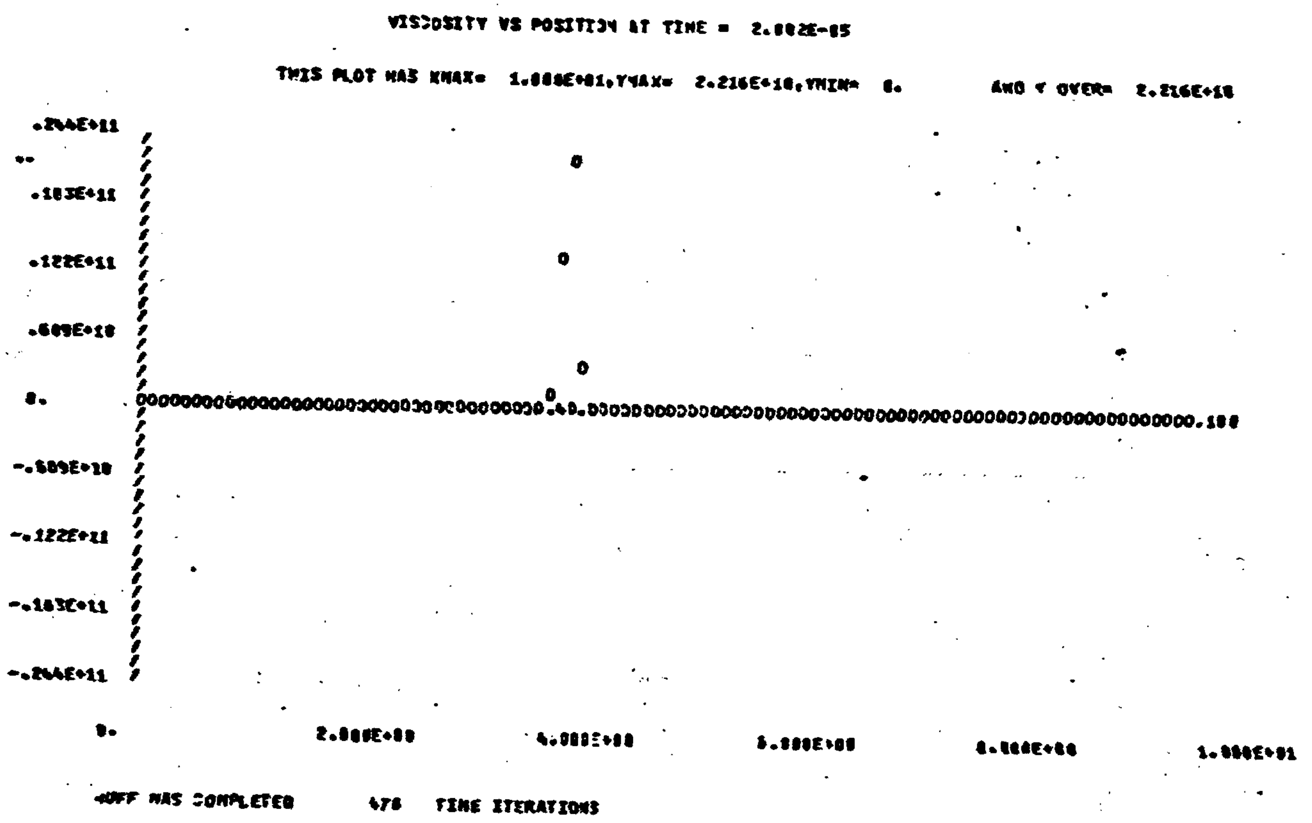


Figure 28. Material Velocity for Spherical Compression of Uranium (2x10^{-5}sec)



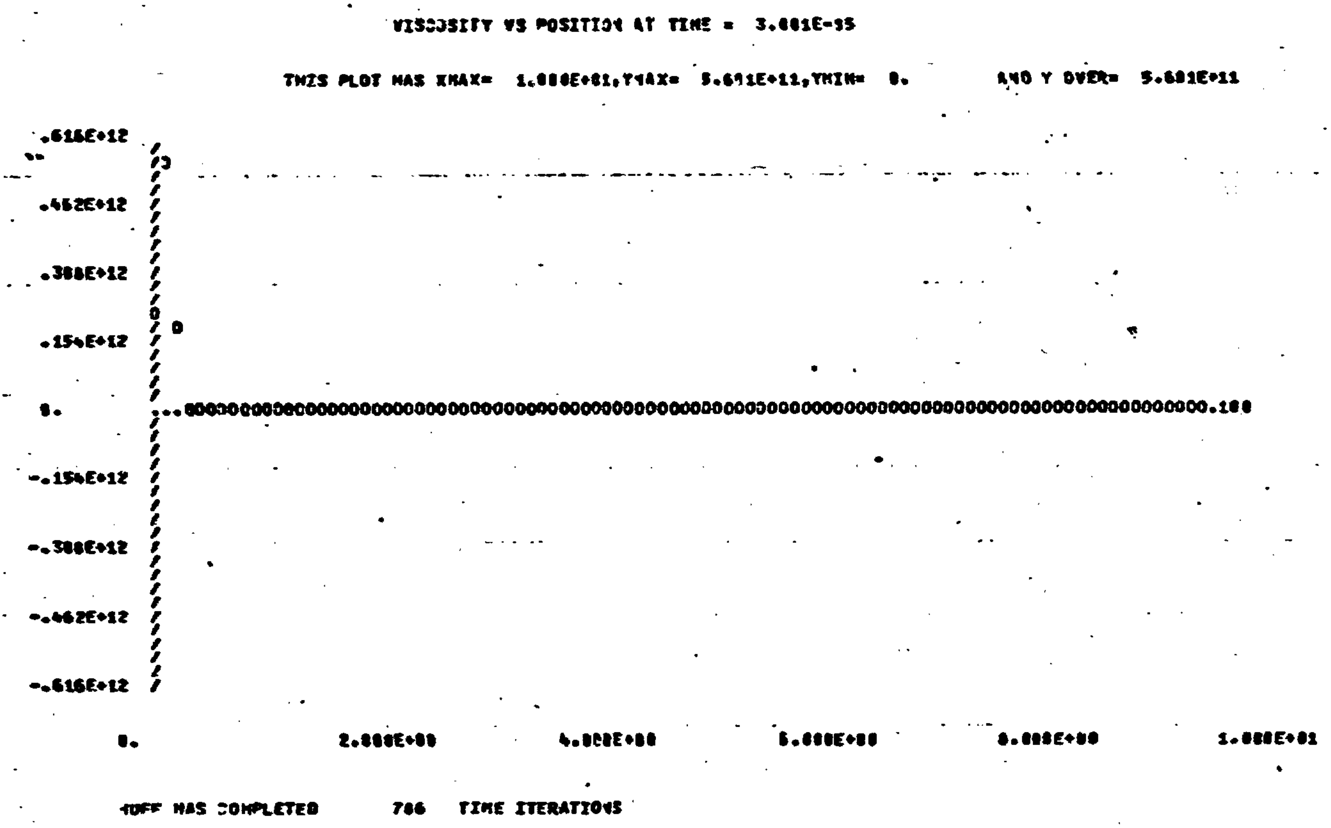
```
VELOCITY VS POSITION AT TIME =  3.001E-05

THIS PLOT HAS XMAX=  1.000E+01,YMAX=  0.        ,YMIN= -1.796E+05 AND Y OVER=  2.056E+03
```

Figure 29.. Material Velocity for Spherical Compression of Uranium ($3x10^{-5}$sec)



```
VISCOSITY VS POSITION AT TIME =  2.002E-05

THIS PLOT HAS XMAX=  1.000E+01,YMAX=  2.216E+10,YMIN=  0.        AND Y OVER=  2.216E+10

PUFF WAS COMPLETED      470   TIME ITERATIONS
```

Figure 30. Artificial Viscosity for Spherical Compression of Uranium (2x10^{-5}sec)



```
VISCOSITY VS POSITION AT TIME =  3.001E-05

THIS PLOT HAS XMAX=  1.000E+01,YMAX=  5.601E+11,YMIN=  0.        AND Y OVER=  5.601E+11

HUFF HAS COMPLETED      786   TIME ITERATIONS
```

Figure 31. Artificial Viscosity for Spherical Compression of Uranium (3x10^{-5}sec)

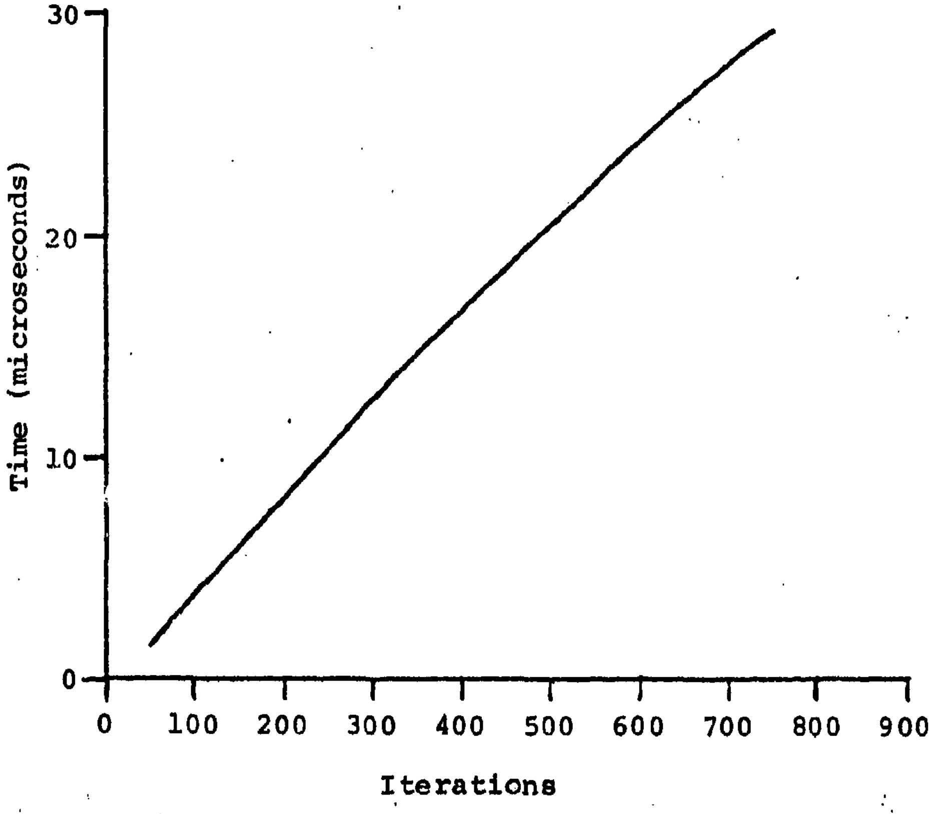


Figure 32. Time vs Iterations for Spherical Compression of Uranium

Appendix D

Derivations

The derivations presented in this appendix develop the key equations used in program HUFF. Coupling of the equation of state with the Richtmyer and Morton difference equation for energy will be presented first. Next, the derivation of the adiabatic sound speed is presented. The last derivation describes the basis for the equations used in rezoning the mesh in the blast expansion problem.

Equation of State Coupling

Hydrodynamic shock propagation results from an imbalance of the interrelated material properties of a gas or solid. Each particle interacts with its adjacent particle in an effort to maintain or reestablish an equalibrium condition in the material. This interaction between particles is simulated in Huff on a macroscopic level by packaging units of material in Lagrangian cells. Each cell interacts with its adjacent cell to maintain or establish equalibrium throughout the mesh.

The solution of the conservation equations of mass, momentum, and energy is dependent on the equation of state of the material. The three equations expressing the conservation laws contain four dependent variables (Ref Equations 1-3). The equation of state defines an additional relation between these variables providing a unique solution.

Solving the conservation equations can be simplified by replacing the expression for energy in the energy conservation equation by its equivalence from the equation of state of the material. This reduces the problem to a solution of three equations and three unknowns and provides for a unique solution.

The difference scheme used in HUFF provides an explicit solution of the conservation equations. The energy conservation equation is expressed as

$$I_j^{n+1} - I_j^n + \left[\frac{P_j^{n+1} + P_j^n}{2} + Q_j^{n+1} \right] (V_j^{n+1} - V_j^n) = 0 \qquad (17)$$

The next two sections eliminate internal energy using the equation of state for a gas and for a solid respectively.

<u>Gas</u>. The equation of state for a gas is

$$P_j^n = (\gamma - 1)\rho_j^n I_j^n \qquad (18)$$

Substituting the expression for I in equation (20) into equation (17) gives

$$\frac{P_j^{n+1}}{(\gamma - 1)\rho_j^{n+1}} - \frac{P_j^n}{(\gamma - 1)\rho_j^n} + \left[\frac{P_j^{n+1} + P_j^n}{2} + Q_j^{n+1} \right] (V_j^{n+1} - V_j^n) = 0 \qquad (42)$$

Multiplying by 2 and combining terms with P and Q gives

$$P_j^{n+1} \left[(V_j^{n+1} - V_j^n) + \left(\frac{2}{(\gamma - 1)}\right) V_j^{n+1} \right] +$$

$$P_j^n \left[(V_j^{n+1} - V_j^n) - \left(\frac{2}{\gamma - 1}\right) V_j^n \right] + Q_j^{n+1} (V_j^{n+1} - V_j^n) = 0 \qquad (43)$$

Solving for P_j^{n+1} gives

$$P_j^{n+1} = \frac{P_j^n \{SV_j^n - (V_j^{n+1} - V_j^n)\} - 2Q_j^{n+1}(V_j^{n+1} - V_j^n)}{SV_j^{n+1} + (V_j^{n+1} - V_j^n)} \tag{21}$$

where $S = \dfrac{2}{\gamma - 1}$.

<u>Solid</u>. The equation of state for a solid is

$$P_j^n = (P_H)_j^n + \gamma_S \rho_j^n \left[I_j^n - (I_H)_j^n \right] \tag{23}$$

$$(P_H)_j^n = \frac{c^2(V_o - V\ )}{\{V_o - S(V_o - V_j^n)\}^2} \tag{24}$$

$$(I_H)_j^n = \frac{1}{2} \left[\frac{c(V_o - V_j^n)}{V_o - S(V_o - V_j^n)} \right]^2 \tag{25}$$

$$S = \frac{(\gamma_S + 1)}{2} \tag{26}$$

Solving for I_j^n in equation (23) gives

$$I_j^n = (I_H)_j^n + \frac{V_j^n}{\gamma_S} \left[P_j^n - (P_H)_j^n \right] \tag{44}$$

Substituting equation (44) into equation (17) gives

$$\left[\frac{V_j^{n+1}}{\gamma_S} P_j^{n+1} - (P_H)_j^{n+1} + (I_H)_j^{n+1}\right] - \left[\frac{V_j^n}{\gamma_S}(P_j^n - (P_H)_j^n) + (I_H)_j^n\right]$$

$$+ \left[\frac{P_j^{n+1} + P_j^n}{2} + Q_j^{n+1}\right](V_j^{n+1} - V_j^n) = 0 \qquad (45)$$

Multiplying by $2\gamma_S$ and collecting terms with P and Q gives

$$P_j^{n+1}\left[2V_j^{n+1} + \gamma_S(V_j^{n+1} + V_j^n)\right] + P_j^n\left[\gamma_S(V_j^{n+1} - V_j^n) - 2V_j^n\right] - 2(P_H)_j^{n+1}V_j^{n+1}$$

$$+ 2(P_H)_j^n V_j^n + 2\gamma_S\left[(I_H)_j^{n+1} - (I_H)_j^n\right] + 2\gamma_S Q_j^{n+1}(V_j^{n+1} - V_j^n) = 0$$

$$(46)$$

Solving for P_j^{n+1} gives

$$P^{n+1} = \frac{P_j^n\{2V_j^n - \gamma_S(V_j^{n+1} - V_j^n)\} - 2\gamma_S\{(I_H)_j^{n+1} - (I_H)_j^n\}}{2V_j^{n+1} + \gamma_S(V_j^{n+1} - V_j^n)}$$

$$+ \frac{2V_j^{n+1}(P_H)_j^{n+1} - 2V_j^n(P_H)_j^n - 2\gamma_S Q_j^{n+1}(V_j^{n+1} - V_j^n)}{2V_j^{n+1} + \gamma_S(V_j^{n+1} - V_j^n)} \qquad (27)$$

__Derivation of Adiabatic Sound Speed__

The adiabatic sound speed is given by

$$(C_A)_j^n = \left(\frac{\partial P}{\partial \rho}\Big|_j^n{}_A\right)^{1/2} \qquad (29)$$

Derivation of the adiabatic sound speed is presented first for a gas and then for a solid.

Gas.

$$P = (\gamma - 1)\rho I \qquad (20)$$

The derivation of pressure with respect to density is given by

$$\frac{\partial P}{\partial \rho} = (\gamma - 1)I + (\gamma - 1)\rho\frac{\partial I}{\partial \rho} \qquad (47)$$

Replacing I in equation (47) with its equivalence from equation (20) gives

$$\frac{\partial P}{\partial \rho} = \frac{P}{\rho} + (\gamma - 1)\rho\frac{\partial I}{\partial \rho} \qquad (48)$$

Under adiabatic conditions $\frac{\partial I}{\partial \rho}$ is given by (Ref 3:5)

$$\frac{\partial I}{\partial \rho} = \frac{P}{\rho^2} \qquad (49)$$

Substituting equation (49) into equation (48) gives

$$\frac{\partial P}{\partial \rho} = \frac{P}{\rho} + (\gamma - 1)\frac{P}{\rho} = \frac{\gamma P}{\rho} \qquad (50)$$

Thus

$$C_A = \left[\frac{\gamma P}{\rho}\right]^{1/2} \qquad (29)$$

Solid. The equation of state for a solid is

$$P = P_H + \frac{\gamma_s}{V}(I - I_H) \qquad (23)$$

The derivative of pressure with respect to density is given by

$$\frac{\partial P}{\partial \rho} = \frac{\partial P_H}{\partial \rho} + \frac{\gamma_S}{V}\left[\frac{\partial I}{\partial \rho} - \frac{\partial I_H}{\partial \rho}\right] + \gamma_S(I - I_H) \tag{51}$$

The expressions for P_H and I_H are

$$P_H = \frac{c^2(V_o - V)}{\{V_o - S(V_o - V)\}^2} \tag{24}$$

$$I_H = \frac{1}{2}\left[\frac{c(V_o - V)}{V_o - S(V_o - V)}\right]^2 \tag{25}$$

Thus the derivatives are

$$\frac{\partial P_H}{\partial \rho} = \frac{c^2V^2\{V_o - S(V_o - V)\} - 2c^2(V_o - V)(-SV^2)}{\{V_o - S(V_o - V)\}^3} \tag{52}$$

$$\frac{\partial I_H}{\partial \rho} = \left[\frac{c(V_o - V)}{V_o - S(V_o - V)}\right]\left[\frac{cV^2\{V_o - S(V_o - V)\} + cSV^2(V_o - V)}{\{V_o - S(V_o - V)\}^2}\right] \tag{53}$$

These expression simplify to the following:

$$\frac{\partial P_H}{\partial \rho} = \frac{c^2V^2V_o(1 + S) - c^2V^3S}{\{V_o - S(V_o - V)\}^3} \tag{54}$$

$$\frac{\partial I_H}{\partial \rho} = \frac{c^2V^2V_o{}^2 - c^2V^3V_o}{\{V_o - S(V_o - V)\}^3} \tag{55}$$

For adiabatic conditions $\frac{\partial I}{\partial \rho}$ is given by

$$\frac{\partial I}{\partial \rho} = \frac{P}{\rho^2} = PV^2 \tag{49}$$

Substituting the above expressions and the expression for I from the equation of state into equation (51) gives

$$\frac{\partial P}{\partial \rho} = \frac{c^2V^2V_o(1 + S) - c^2V^2S}{\{V_o - S(V_o - V)\}^3} + \frac{\gamma_S}{V}\left[PV^2 - \left(\frac{c^2V^2V_o^2 - c^2V^3V_o}{\{V_o - S(V_o - V)\}^3}\right)\right]$$

$$+ \gamma_S\left[\frac{P - P_H}{\gamma_s \rho} + I_H - I_H\right] \tag{56}$$

Substituting the value for P_H into equation (56), collecting terms and simplifying gives

$$\frac{\partial P}{\partial \rho} = \frac{c^2V^2V_o(1 + S + \gamma_S) - c^2V^3S - \gamma_S c^2VV_o{}^2}{\{V_o - S(V_o - V)\}^3}$$

$$- \frac{c^2(V_o - V)V}{\{V_o - S(V_o - V)\}^2} + PV(\gamma_s + 1) \tag{57}$$

Substituting $\gamma_S = 2S - 1$ into equation (57), combining terms and simplifying gives

$$\frac{\partial P}{\partial \rho} = 2SVP + \frac{2Sc^2V^2V_o - Sc^2VV_o{}^2}{\{V_o - S(V_o - V)\}^3} \tag{58}$$

Thus the adiabatic sound speed in solids is given by

$$C_A = \left[2SVP + \frac{2Sc^2V^2V_o - Sc^2VV_o^2}{V_o - S(V_o - V)^3}\right]^{1/2} \qquad (30)$$

Rezone Computations from Conservation Requirements

Rezoning during the blast expansion problem provides the capability to use small mesh spacing during the early phases of shock development and also follow the shock to great distances later in the problem. Rezoning is accomplished when the shock reaches the 94th cell of the mesh.

Rezoning compresses the information stored in the 100 cell mesh into the first 50 cells and places undisturbed air in the last 50 cells. The rezone process carefully conserves mass, momentum, and total energy to preserve the material imbalance present in the mesh prior to rezoning.

This section presents the derivations that provide the equations used in rezoning. The derivation begins with defining the location of the new cells and proceeds in the order of solution used in subroutine REZONE.

Position. The new cell locations in the mesh are formed by removing the even numbered cell boundaries and combining the adjacent cells. The old cell positions use the subscript j and the new cell positions use the subscript i . The location of each new cell boundary is given by

$$x_i = \frac{x_j + 1}{2} \qquad (60)$$

where j are the even cell boundaries from 2 to 98. Cell boundary locations after cell 49 are computed from the previous cell position by

$$X_i = X_{i-1} + DX \qquad (61)$$

where DX is twice the original cell spacing existing before the rezone.

Conservation of Mass. Conservation of mass provides an explicit solution of the new cell density and mass. The equation for conservation of mass is

$$P_i(X^3_{i+1} - X^3_i) = P_j(X^3_{j+1} - X^3_j) + P_{j+1}(X^3_{j+2} - X^3_{j+1}) \qquad (62)$$

The expression for density is

$$P_i = \frac{P_j(X^3_{j+1} - X^3_j) + P_{j+1}(X^3_{j+2} - X^3_{j+1})}{(X^3_{j+2} - X^3_j)} \qquad (63)$$

where $X_{j+2} = X_{i+1}$ and $X_j = X_i$.

Cell Centered Velocity. The time step calculations accomplished by program HUFF use the cell boundary velocities to change the cell positions. To conserve momentum and energy in the cells during rezone, the average velocity of each cell is required. The average cell velocity is computed by averaging the cell boundary velocities assuming the velocity varies linearly across the cell. The average cell velocity is given by

$$u_j = \frac{u_j + u_{j+1}}{2} \qquad (64)$$

where $\bar{u}_j$ is the cell centered velocity for the j^{th} cell.

Total Energy. The total energy of each cell is computed from the cell centered velocity and the equation of state. The equation for total energy is given by

$$E_j = \frac{(\bar{u}_j)^2}{2} + \frac{P_j}{(\gamma - 1)\rho_j} \qquad (65)$$

where E_j is the total energy of cell j and a variable gamma is used computed from the pressure of cell j .

Conservation of Energy. Conservation of energy provides an explicit solution for the total energy in the new cell. The equation for conservation of energy is

$$\rho_i(X^3_{i+1} - X^3_i)E_i = \rho_j(X^3_{j+1} - X^3_j)E_j + \rho_{j+1}(X^3_{j+2} - X^3_{j+1})E_{j+1} \qquad (66)$$

The expression for the new cell energy is

$$E_i = \frac{\rho_j(X^3_{j+1} - X^3_j)E_j + \rho_{j+1}(X^3_{j+2} - X^3_{j+1})}{\rho_i(X^3_{j+2} - X^3_j)} \qquad (67)$$

Conservation of Momentum. Conservation of momentum provides an explicit solution for the cell centered velocity of the new cell. The equation for conservation of momentum is

$$\rho_i(X^3_{i+1} - X^3_i)\bar{u}_i = \rho_j(X^3_{j+1} - X^3_j)\bar{u}_j + \rho_{j+1}(X^3_{j+2} - X^3_{j+1})\bar{u}_{j+1} \qquad (68)$$

The expression for the cell centered velocity of the new cell is

$$\bar{u}_i = \frac{\rho_j(X^3_{j+1} - X^3_j)\bar{u}_j + \rho_{j+1}(X^3_{j+2} - X^3_{j+1})\bar{u}_{j+1}}{\rho_i(X^3_{j+2} - X^3_j)} \qquad (69)$$

Cell Boundary Velocities. The new cell boundary velocities are recomputed using a procedure similar to that used in computing the cell centered velocities. The cell boundary velocity is computed by averaging the cell centered velocities of the adjacent cells assuming the velocity varies between the adjacent cell centers in a manner that gives an average velocity at the common boundary. The expression for the cell boundary velocity of the new cell is given by

$$u_i = \frac{\bar{u}_{i-1} + \bar{u}_i}{2} \tag{70}$$

The cell boundary velocity at the center position, u_1, is defined as zero and is not computed by equation (70).

Internal Energy. The internal energy of the new cell is calculated by the application of the conservation of total energy. The kenetic energy of the new cell is calculated first and is given by

$$KE_i = \frac{\bar{u}_i}{2} \tag{71}$$

where KE_i is the kenetic energy of the new cell i. Total energy conservation expressed in its most simple form is

$$E_i = KE_i + I_i \tag{72}$$

where I_i is the internal energy of the new cell.

The expression for the internal energy of the new cell is given by:

$$I_i = E_i - KE_i \tag{73}$$

Pressure. Pressure is computed using the equation of state. The expression for pressure is

$$P_i = (\gamma - 1)\rho_i I_i \tag{20}$$

where $(\gamma - 1)$ is a function of internal energy. The value of $(\gamma - 1)$ is computed directly usinj the internal energy already computed for the new cell.

Computing the pressure completes the rezoning process and the rezoned mesh is returned to subroutine CONTROL to continue the shock propagation. The mesh will again return to subroutine REZONE when the shock position again reaches the 94th cell.

Appendix E

Program Listing of HUFF

```
      PROGRAM HUFF(INPUT=/80,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT
     1,FILES,REVIEW,TAPE7=FILES,TAPE8=REVIEW)
      DIMENSION FLAG(4),TABLE(10)
      INTEGER FLAG
      CALL FLAGS(FLAG,TABLE,TSTOP)
      CALL CONTROL(FLAG,TABLE,TSTOP)
      PRINT*,"TIME LIMIT EXCEEDED"
      STOP
      END
C
      SUBROUTINE FLAGS(FLAG,TABLE,TSTOP)
      DIMENSION FLAG(4),TABLE(10)
      INTEGER FLAG
C----- INITIALIZE CONTROL PARAMETERS ---------------------------------
      DO 10 I=1,3
   10 FLAG(I)=0
      DO 30 I=1,10
   30 TABLE(I)=0
C      I READ FLAGS AND REAL TIME LIMIT              I
C------I      FLAG(2)=# OF TABLES REQUESTED         I-------------
C      I      FLAG(3)=# OF CYCLES THROUGH HYDRO     I
C      I      FLAG(4)=TYPE PROBLEM (1=BLAST,2=SQUEEZE)I
      READ*,(FLAG(I),I=2,4),TSTOP
C------ READ REQUESTED TABLE TIMES -------------------------
      IF2=FLAG(2)
      READ*,(TABLE(I),I=1,IF2)
C------WRITE OUT INFORMATION PROVIDED BY FLAGS --------------
          IF(FLAG(4).EQ.1)WRITE(2,400)
          IF(FLAG(4).EQ.2)WRITE(2,500)
          IF(FLAG(4).NE.1.AND.FLAG(4).NE.2)WRITE(2,600)FLAG(4)
      WRITE(2,200)FLAG(2),(TABLE(I),I=1,IF2)
      WRITE(2,300)TSTOP,FLAG(3)
  200 FORMAT(////,T10,"TABLES FOR THE FOLLOWING ",I3,
     1" TIMES WERE REQUESTED:",//,2(T15,5(1PE10.3,2X),/))
  300 FORMAT(////,T10,"MAX TIME TO RUN IS ",F4.1," SEC."
     1,/,T10,"MAX CYCLES THROUGH TIMESTEP IS",I9,".")
  400 FORMAT(1H1,T40,"THIS IS A BLAST PROBLEM")
  500 FORMAT(1H1,T40,"THIS IS A SQUEEZE PROBLEM")
  600 FORMAT(1H1,T40,"NO SUCH PROBLEM NUMBER = ",I3)
      RETURN
      END
```

```
520=      SUBROUTINE CONTROL(FLAG,TABLE,TSTOP)
530=      DIMENSION X(101),U(101),P(101),D(101),Q(101),FLAG(4)
540=     1,TABLE(10),CK(7)
550=      INTEGER FLAG
560=C------CALL TO SET INITIAL CONDITIONS ------------------------------
570=          IF(FLAG(4).EQ.1)
580=     1CALL BLAST(CA,IALPHA,GAMMA,N1,M,DX,RHO,P,D,X,Q,U,DT,CK)
590=          IF(FLAG(4).EQ.2)
600=     1 CALL SQUEEZE(CA,IALPHA,GAMMA,N1,DX,RHO,P,D,X,Q,U,C2,DT,CK)
610=         N1=0
620=C------LIST INITIAL CONDITIONS -------------------------------------
630=      TIME=CK(2)
640=      CALL ICOPY(X,U,P,D,Q,FLAG(2),TIME,TABLE,CK)
650=      IF3=FLAG(3)
660=      DO 15 ICYCLE=1,IF3
670=C------CALL TO ADVANCE ONE TIMESTEP --------------------------------
680=       CALL HYDRO(D,P,X,Q,U,DX,CA,IALPHA,RHO,GAMMA,N1,DT,TIME,FLAG,C2)
690=C------CALL TO SET BOUNDARY CONDITIONS AND TRACK SHOCK -------------
700=          IF(FLAG(4).EQ.1)
710=     1CALL EXPAND(CA,IALPHA,GAMMA,N1,M,DX,RHO,P,D,X,Q,U,DT,CK)
720=          IF(FLAG(4).EQ.2)
730=     1 CALL CRUSH(CA,IALPHA,GAMMA,N1,DX,RHO,P,D,X,Q,U,C2,DT,CK)
740=      TIME=CK(2)
750=C------ CALL TO COPY REQUESTED TABLES --------------------
760=       IF(TIME.GE.TSTOP.OR.TIME.GE.TABLE(1))
770=     1CALL COPY(X,U,P,D,Q,FLAG(2),TIME,TABLE,CK)
780=C------ ABORT IF TIME LIMIT REACHED --------------------------------
790=       IF(TIME.GE.TSTOP)RETURN
800=C------ CALL FOR REZONE IF REQUIRED --------------------------------
810=      IF(N1.LT.94.OR.FLAG(4).NE.1)GO TO 15
820=      CALL REZONE(X,U,P,D,Q,DX,GAMMA,ICYCLE,TIME,N1,RHO)
830=   15 CONTINUE
840=C------ CALL RESULTS AND ABORT PROGRAM IF CYCLE LIMIT REACHED -------
850=      CALL COPY(X,U,P,D,Q,FLAG(2),TIME,TABLE,CK)
860=      PRINT*,"YOU REACHED THE CYCLE LIMIT OF ",ICYCLE
870=      PRINT*,"TIME IS ",TIME," SEC."
880=      RETURN
890=      END
900=C
910=      SUBROUTINE HYDRO(D,P,X,Q,U,DX,CA,L,RHO,GAMMA,N1,DT,TIME,FLAG,C2)
920=      DIMENSION D(101),P(101),X(101),Q(101),U(101),FLAG(4)
930=     1,VC(101),VEL(101)
940=      INTEGER FLAG
950=C------ INITIALIZE LOCAL PARAMETERS -------------------------
960=      VEL(1)=0
970=      DT=1.
980=      M=N1+5
990=      IF(FLAG(4).EQ.2)M=100
1000=     IF(M.LT.15)M=15
```

```
C-------- INITIALIZE HYDROMATIC TIME STEP --------------------
      IF(N1.NE.0)GO TO 10
      DO 5 I=1,101
    5 VG(I)=1.4
       TFRAC=.1
       IF(FLAG(4).EQ.2)N1=95
       IF(FLAG(4).EQ.1)N1=10
       ICOUNT=0
   10 ICOUNT=ICOUNT+1
      NSAVE=N1
   15 CONTINUE
      DO 20 J=2,M
      IF(FLAG(4).EQ.2)GO TO 17
C------ COMPUTE GAS TIMESTEP PARAMETER -----------------------
      A=D(J)/(VG(J)*P(J))
      GO TO 18
   17 CONTINUE
C-------COMPUTE SOLID TIMESTEP PARAMETER --------------------
      V=1./D(J)
      VRHO=1./RHO
      S=(GAMMA+1.)/2.
      A=1./(2.*S*P(J)*V+(C2*V**2*VRHO*(1.+S)-S*
     1C2*V*VRHO**2)/(VRHO-S*(VRHO-V))**3)
   18 CONTINUE
C------ COMPUTE LOCAL TIMESTEP ------------------------------
      IF(A.LE.0.)GO TO 20
       DTJ=(X(J+1)-X(J))*TFRAC*SQRT(A)
C------- LOCATE MINIMUM TIMESTEP ----------------------------
       IF(DTJ.LT.DT)DT=DTJ
   20 CONTINUE
      DO 25 I=2,M
C------- COMPUTE NEW CELL BOUNDARY VELOCITIES ----------------
      XI=I
       VEL(I)=U(I)+DT*(P(I-1)-P(I)+Q(I-1)-Q(I))/
     1(RHO*DX)*(X(I)/((XI-1)*DX))**(L-1)
C------ CHECK CONVERGENCE VELOCITY OF CELL BOUNDARIES --------
       IF((VEL(I-1)-VEL(I))*DT.LT..5*(X(I)-X(I-1)))GO TO 25
        TFRAC=TFRAC*.5
      PRINT*," CONVERGENCE PROBLEM;TFRAC=",TFRAC
        GO TO 15
   25 CONTINUE
      DO 30 I=2,M
      U(I)=VEL(I)
C------- COMPUTE NEW CELL BOUNDARY POSITIONS -----------------
       X(I)=X(I)+DT*U(I)
   30 CONTINUE
```

```
C------- COMPUTE NEW DENSITIES AND SPECIFIC VOLUMES ----------
      VRHO=1./RHO
       QMAX=10.
      DO 40 I=1,M
       XI=I
       VOLD=1./D(I)
      D(I)=RHO*((XI*DX)**L-((XI-1.)*DX)**L)/
     1(X(I+1)**L-X(I)**L)
       VNEW=1./D(I)
      VDOT=VNEW-VOLD
C------- COMPUTE NEW VISCOSITIES ----------------------------
       Q(I)=2.*CA/(VNEW+VOLD)*(U(I+1)-U(I))**2
       IF(U(I+1)-U(I).GE.0.)Q(I)=0.
      IF(FLAG(4).EQ.1)GO TO 38
C------ COMPUTE PRESSURES FOR SOLID -------------------------
      S=(GAMMA+1.)/2.
      VSVOLD=VRHO-S*(VRHO-VOLD)
      VSVNEW=VRHO-S*(VRHO-VNEW)
      HPOLD=C2*(VRHO-VOLD)/VSVOLD**2
      HPNEW=C2*(VRHO-VNEW)/VSVNEW**2
      HIOLD=C2*((VRHO-VOLD)/VSVOLD)**2/2.
      HINEW=C2*((VRHO-VNEW)/VSVNEW)**2/2.
      P(I)=(P(I)*(2.*VOLD-GAMMA*VDOT)-2.*GAMMA*(HINEW-HIOLD)+
     12.*(VNEW*HPNEW-VOLD*HPOLD)-2.*GAMMA*Q(I)*VDOT)/
     2(2.*VNEW+GAMMA*VDOT)
      GO TO 40
   38 CONTINUE
C------ COMPUTE PRESSURE USING VARIABLE GAMMA ---------------
      S=2./(VG(I)-1.)
      PX=(P(I)*(S*VOLD-VDOT)-2.*Q(I)*VDOT)/(S*VNEW+VDOT)
      EI=EIA(PX,D(I),RHO)
      VG(I)=VGAMMA(EI,D(I),RHO)
      S=2./(VG(I)-1.)
      P(I)=(P(I)*(S*VOLD-VDOT)-2.*Q(I)*VDOT)/(S*VNEW+VDOT)
   40 CONTINUE
C------ LOCATE NEW SHOCK POSITION ---------------------------
      NP5=N1+5
      NM5=N1-5
      DO 52 I=NM5,NP5
      IF(QMAX.GT.Q(I))GO TO 52
      QMAX=Q(I)
      N1=I
   52 CONTINUE
C------ ADVANCE TIME ----------------------------------------
      TIME=TIME+DT
C-------- HYDROMATIC TIME STEP CONTROL ---------------------
      IF(NSAVE-N1.EQ.0)RETURN
      IF(ICOUNT.LE.5)TFRAC=.5*TFRAC
      IF(ICOUNT.GT.10)TFRAC=TFRAC+.1
      ICOUNT=0
      RETURN
      END

      SUBROUTINE SQUEEZE(CA,L,GAMMA,N1,DX,RHO,P,D,X,Q,U,C2,DT,CK)
      DIMENSION P(101),D(101),X(101),Q(101),U(101),CK(7)
      CA=4
C------ READ PROBLEM GEOMETRY,GAMMA,DENSITY,SOUND SPEED,
C         RADIUS OF METEL, &ND MASS OF HIGH EXPLOSIVE ----------
      READ*,L,GAMMA,RHO,C,RADIUS,PRESS
      WRITE*,L,GAMMA,RHO,C,RADIUS,PRESS
C------ COMPUTE INITIAL CONDITIONS FOR PROBLEM ---------------
      DX=RADIUS/100.
      C2=C**2
      PRINT*,"DX=",DX," AND C2=",C2
      PNORM=1.014E6
C------ ASSIGN SCALED VALUES TO PROBLEM ARRAYS --------------
      DO 10 I=1,101
       U(I)=Q(I)=0
       P(I)=PNORM
       D(I)=RHO
       IF(I.EQ.1)X(I)=0
       IF(I.GT.1)X(I)=X(I-1)+DX
   10 CONTINUE
C------ INITIALIZE LOCAL PARAMETERS -------------------------
      NN=N1=100
      OPOS=X(N1)
      TIMIT=CYCLE=KFIFTY=0
      P(100)=PRESS
      RETURN
C
      ENTRY CRUSH
C------- INPUT A CARD HERE FOR CONST APPLIED PRESSURE -------
C----- CARD SHOULD READ (P(100)=PRESS) ---------------------
      IF(P(100).LT.PNORM)P(100)=PNORM
C------ RESET SEMETRIC B. C. AND ADVANCE TIME ---------------
      X(1)=U(1)=0.
C------ TIME SHOCK PROGRESS FOR 50 ITERATIONS ---------------
      KFIFTY=KFIFTY+1
      TIMIT=TIMIT+DT
C------ COMPUTE OVERPRESSURE AND OVERDENSITY ----------------
C------ PLACE COMPUTED VALUES IN CHECK (CK) ARRAY ------------
       IP=N1
      CK(1)=CYCLE=CYCLE+1.
      CK(2)=TIME=TIME+DT
      CK(3)=DT
      CK(4)=POS=X(IP)
      PMAX=DMAX=VMAX=0.
      NM5=N1-5
      NP5=N1+5
      DO 20 I=NM5,NP5
      IF(VMAX.LT.U(I))VMAX=U(I)
      IF(P(I).GT.PMAX)PMAX=P(I)
      IF(D(I).GT.DMAX)DMAX=D(I)
   20 CONTINUE
```

```
      CK(5)=OP=PMAX-PNORM
      CK(6)=OD=DMAX-RHO
C------ COMPUTE NEW MATERIAL VELOCITY ---------------------
      CK(7)=VMAX
      IF(KFIFTY.LT.50)GO TO 40
      OPOS=POS
      TIMIT=KFIFTY=0
C------ WRITE CHECK VALUES EVERY 50 ITERATIONS ---------------
      WRITE(8,2000)(CK(I),I=1,7)
   40 CONTINUE
  110 FORMAT(/,10X,"CORE IS SOLID WITH ",G10.2," GMS HE.",/)
 2000 FORMAT(1X,"CYCLE=",F5.0," T(SEC)=",1PE9.2," DT(SEC)="
     1,E7.0," POS(CM)=",E9.2," OP(D/CM**2)=",E9.2," OD(G/CM**3)="
     2,E9.2," MVEL(CM/SEC)=",E9.2)
      RETURN
      END

      SUBROUTINE BLAST(CA,L,GAMMA,N1,M,DX,RHO,P,D,X,Q,U,DT,CK)
      DIMENSION P(101),D(101),X(101),Q(101),U(101),CK(7)
      SCALE(XNORM,XD)=XNORM*YKT**(1./3.)*XD
      CA=4
C------ READ YIELD,SCALING PARAMETERS,AND PROBLEM GEOMETRY ----
      READ*,YKT,SP,SD,ST
      L=3
      WRITE*,YKT,SP,SD,ST,L
C------ COMPUTE INITIAL CONDITIONS USING SCALING LAWS --------
      PNORM=1.014E6*SP
      CK(2)=TIME=SCALE(3.68E-5,ST)
      PBLAST=1.5E10*SP
      DX=SCALE(50.,SD)
      RHO=1.293E-3*SD
      N1=9
      OPOS=X(N1)
      PRINT*,"PNORM=",PNORM," PBLAST=",PBLAST
      PRINT*," TIME=",TIME," DX=",DX," RHO=",RHO," N1=",N1
C------ ASSIGN SCALED VALUES TO PROBLEM ARRAYS ---------------
      DO 10 I=1,101
       Q(I)=U(I)=0
       D(I)=RHO
       IF(I.EQ.1)X(I)=0
       IF(I.GT.1)X(I)=X(I-1)+DX
       IF(I.LE.N1)P(I)=PBLAST
       IF(I.GT.N1)P(I)=PNORM
   10 CONTINUE
C------ INITIALIZE LOCAL PARAMETERS --------------------------
      M=N1+5
      TIMIT=CYCLE=KFIFTY=0
      RETURN

      ENTRY EXPAND
C------ RESET SEMETRIC B. C. AND ADVANCE TIME ----------------
      X(1)=U(1)=0.
C------ LOCATE NEW SHOCK POSITION ----------------------------
      IP=N1
C------ TIME SHOCK PROGRESS FOR 50 ITERATIONS ----------------
      KFIFTY=KFIFTY+1
      TIMIT=TIMIT+DT
C------ COMPUTE OVERPRESSURE AND OVERDENSITY -----------------
      PMAX=DMAX=VMAX=0.
      NM5=N1-5
      NP5=N1+5
      DO 20 I=NM5,NP5
      IF(U(I).GT.VMAX)VMAX=U(I)
      IF(P(I).GT.PMAX)PMAX=P(I)
      IF(D(I).GT.DMAX)DMAX=D(I)
   20 CONTINUE
      CK(5)=OP=PMAX-PNORM
      CK(6)=OD=DMAX-RHO
C------ PLACE COMPUTED VALUES IN CHECK (CK) ARRAY ------------
      CK(1)=CYCLE=CYCLE+1.
      CK(2)=TIME=TIME+DT
      CK(3)=DT
      CK(4)=POS=X(IP)
C------ COMPUTE NEW SHOCK MATERIAL VELOCITY ---------------
      CK(7)=VMAX
      IF(KFIFTY.LT.50)GO TO 30
      OPOS=POS
      TIMIT=KFIFTY=0
C------ WRITE CHECK VALUES EVERY 50 ITERATIONS ---------------
      WRITE(8,2000)(CK(I),I=1,7)
      WRITE(2,100)(CK(K),K=1,7)
   30 M=N1+5
  100 FORMAT(2X,"CYCLE=",F6.0," TIME=",1PE9.3," DT=",E9.3,
     1" S POS =",E9.3," O PRES=",E9.3," O DEN="E9.3," S VEL="
     2,E9.3)
 2000 FORMAT(1X,"CYCLE=",F5.0," T(SEC)=",1PE9.2," DT(SEC)="
     1,E7.0," POS(CM)=",E9.2," OP(D/CM**2)=",E9.2," OD(G/CM**3)="
     2,E9.2," MVEL(CM/SEC)=",E9.2)
      RETURN
      END
C
      SUBROUTINE REZONE(X,U,P,D,Q,DX,GAMMA,ICYCLE,TIME,N1,RHO)
      DIMENSION X(101),U(101),P(101),D(101),PUD(101,3),Q(101)
      N1=N1/2
      WRITE(2,100)TIME,ICYCLE
      DO 10 J=3,97,2
       I=(J+1)/2
```

```
C ------- COMPUTE GAMMAS ----------------------------------
        EIJ1=EIA(P(J),D(J),RHO)
        EIJ2=EIA(P(J+1),D(J+1),RHO)
        AGAMA=VGAMMA(EIJ1,D(J),RHO)
        BGAMA=VGAMMA(EIJ2,D(J+1),RHO)
C-------- COMPUTE MASS AND NEW DENSITY --------------------
       XMJ1=D(J)*(X(J+1)**3-X(J)**3)
       XMJ2=D(J+1)*(X(J+2)**3-X(J+1)**3)
       PUD(I,3)=(XMJ1+XMJ2)/(X(J+2)**3-X(J)**3)
       XMI1=PUD(I,3)*(X(J+2)**3-X(J)**3)
C--------  COMPUTE OLD CELL CENTERED VELOCITY --------------
       UJ1=(U(J)+U(J+1))/2.
       UJ2=(U(J+1)+U(J+2))/2.
C-------- COMPUTE OLD ENERGIES (KENETIC, INTERNAL, TOTAL) ---
       EKJ1=UJ1**2/2.
       EKJ2=UJ2**2/2.
       EIJ1=P(J)/((AGAMA-1.)*D(J))
       EIJ2=P(J+1)/((BGAMA-1.)*D(J+1))
       ETJ1=EKJ1+EIJ1
       ETJ2=EKJ2+EIJ2
C--------- COMPUTE NEW TOTAL ENERGY ------------------------
       ETI1=(XMJ1*ETJ1+XMJ2*ETJ2)/XMI1
C-------- COMPUTE MOMENTUM ---------------------------------
       XMOMJ1=XMJ1*UJ1
       XMOMJ2=XMJ2*UJ2
       PUD(I,2)=(XMOMJ1+XMOMJ2)/XMI1
       XMOMI1=XMI1*PUD(I,2)
C-------- COMPUTE NEW KENETIC AND INTERNAL ENERGY ----------
       EKI1=PUD(I,2)**2/2.
       EII1=ETI1-EKI1
C-------- COMPUTE NEW PRESSURE ------------------------------
       GAMMA=VGAMMA(EII1,PUD(I,3),RHO)
       PUD(I,1)=EII1*(GAMMA-1.)*PUD(I,3)
C------ BYPASS LENGTHY REZONE PRINTOUT ---------------------------------
      IF(N1.GT.10)GO TO 10
       WRITE(2,200)X(J),UJ1,P(J),D(J),XMJ1,XMOMJ1,ETJ1,EKJ1,EIJ1
       WRITE(2,200)X(J+1),UJ2,P(J+1),D(J+1),XMJ2,XMOMJ2,ETJ2,EKJ2,EIJ2
       WRITE(2,250)X(J),UJ1,PUD(I,1),PUD(I,3),XMI1,XMOMI1,ETI1,EKI1,EII1
   10 CONTINUE
C-------- ASSIGN VELOCITY FOR CENTER AND RESET DX ----------
      PUD(1,2)=0
      DX=2.*DX
C-------- STORE NEW VALUES IN PROGRAM ARRAYS ---------------
      DO 20 I=1,49
       J=2*I-1
       X(I)=X(J)
       P(I)=PUD(I,1)
       D(I)=PUD(I,3)
      IF(I.EQ.1)GOTO 20
```

```
C--------- COMPUTE NEW BOUNDARY VELOCITIES ----------------
      U(I)=(PUD(I-1,2)+PUD(I,2))/2.
   20 CONTINUE
C-------- DEFINE VALUES FOR UPPER HALF OF ARRAYS -----------
      DO 25 I=50,101
      X(I)=X(I-1)+DX
      U(I)=Q(I)=0
      P(I)=P(101)
      D(I)=D(101)
   25 CONTINUE
C------- COMPUTE NEW VISCOSITY -----------------------------
      DO 30 I=1,50
      Q(I)=8.*D(I)*(U(I+1)-U(I))**2
      IF(U(I+1)-U(I).GE.0)Q(I)=0
   30 CONTINUE
C-------  SET BOUNDARY CONDITIONS ------------------------
      U(1)=X(1)=0
      P(1)=P(2)
      Q(1)=Q(2)
      D(1)=D(2)
  100 FORMAT(1H1,T20,"REZONE TIME = ",G9.3," ITERATION NUMBER ",I7,
     1////,T4,"POSITION",T18,"VELOCITY",T32,"PRESSURE",T46,"DENSITY",
     2T62,"MASS",T74,"MOMENTUM",T88,"T ENERGY",T102,"K ENERGY",
     3T116,"I ENERGY",/,T7,"CM",T19,"CM/SEC",T33,"D/CM**3",
     4T46,"G/CM**3",T63,"GM",T76,"G-SEC",T91,"ERG/G",T105,"ERG/G",
     5T119,"ERG/G",///)
  200 FORMAT(3X,9(G9.3,5X))
  250 FORMAT(5X,9(G9.3,5X),/)
      RETURN
      END

      FUNCTION EIA(P,D,RHO)
CC------- QUESS GAMMA MINUS ONE ---------------------------
      GM1=.28
      DO 30 I=1,10
C-------- COMPUTE AN INTERNAL ENERGY ----------------------
      EI=P/(GM1*D)
C--------- COMPUTE A NEW GAMMA MINUS ONE ------------------
      GM1=VGAMMA(EI,D,RHO)-1.
C-------- COMPUTE A PRESSURE ------------------------------
      P2=GM1*D*EI
C-------- COMPARE PRESSURES -------------------------------
      IF(ABS(P-P2).LE..001*P)GO TO 40
C-------- GAMMA IS CORRECT WHEN THE PRESSURES AGREE -------
   30 CONTINUE
      PRINT*,"I=",I
   40 EIA=EI
      RETURN
      END
      FUNCTION VGAMMA(EI,D,RHO)
C-------- HUFF ROUGH VARIABLE GAMMA COMPUTATION -----------
      EIE=EI/1.E10
      IF(EIE.GT..1315)GO TO 10
      GMONE=.3981
      GO TO 30
   10 CONTINUE
      IF(EIE.GT.1.)GO TO 20
      GMONE=-.0399*(EIE-.1315)**2+.3981
      GO TO 30
   20 CONTINUE
      GMONE=.16+.624/(2.+EIE)
C-------- ADJUSTMENT FOR DENSITY -------------------------
      RHOZ=D/RHO
      EL=ALOG10(EIE)
      ALPHA=.0577*EL-.0218*EL**2+.0035*EL**3+.0002*EL**4
      GMONE=GMONE*RHOZ**ALPHA
   30 VGAMMA=GMONE+1.
      RETURN
      END

      SUBROUTINE COPY(X,U,P,D,Q,FLAG,TIME,TABLE,CK)
      DIMENSION X(101),U(101),P(101),D(101),Q(101),TABLE(10)
     1,CK(7)
      INTEGER FLAG
C------- RESET CALLING PARAMETERS -------------------------
      FLAG=FLAG-1
      DO 10 I=1,9
   10 TABLE(I)=TABLE(I+1)
      IF(FLAG.LT.0)TABLE(1)=1000.
C-------- WRITE INFORMATION ON LOCAL FILE (FILES) ----------
      ENTRY ICOPY
      WRITE(7,1000)TIME,(X(I),U(I),P(I),D(I),Q(I),I=1,101)
      WRITE(7,2000)(CK(K),K=1,7)
      WRITE(2,1010)TIME
      CALL PLOTSM(X,P,100)
      WRITE(2,2000)(CK(K),K=1,7)
      WRITE(2,1020)TIME
      CALL PLOTSM(X,D,100)
      WRITE(2,2000)(CK(K),K=1,7)
      WRITE(2,1001)TIME
      CALL PLOTSM(X,U,100)
      WRITE(2,2000)(CK(K),K=1,7)
      WRITE(2,1030)TIME
      CALL PLOTSM(X,Q,100)
      WRITE(2,2000)(CK(K),K=1,7)
```

```
4930= 1000 FORMAT(1X,"TIME IS",F9.5,"SEC",//,T7,"POSITION",T22
4940=     1,"VELOCITY",T37,"PRESSURE",T52,"DENSITY",T67,"VISCOSITY"
4950=     2,/,T9,"(CM)",T22,"(CM/SEC)",T37,"(D/CM**2)",T51,"(G/CM**3)"
4960=     3,T67,"(D/CM**2)",//,101(5(5X,1PE10.3),/))
4970= 1001 FORMAT(1H1,T40,"VELOCITY(CM/SEC) VS POSITION(CM) AT TIME = "
4980=     1,1PE10.3," SEC")
4990= 1010 FORMAT(1H1,T40,"PRESSURE(D/CM**2) VS POSITION(CM) AT TIME = "
5000=     1,1PE10.3," SEC")
5010= 1020 FORMAT(1H1,T40,"DENSITY(G/CM**3) VS POSITION(CM) AT TIME = ",
5020=     11PE10.3," SEC")
5030= 1030 FORMAT(1H1,T40,"VISCOSITY(D/CM**2) VS POSITION(CM) AT TIME = "
5040=     1,1PE10.3," SEC")
5050= 2000 FORMAT(1X,"CYCLE=",F5.0," T(SEC)=",1PE9.2," DT(SEC)="
5060=     1,E7.0," POS(CM)=",E9.2," OP(D/CM**2)=",E

5060=     1,E7.0," POS(CM)=",E9.2," OP(D/CM**2)=",E9.2," OD(G/CM**3)="
5070=     2,E9.2," MVEL(CM/SEC)=",E9.2)
5080=      RETURN
5090=      END
5100=
5110=      SUBROUTINE PLOTSM(X,Y,N)
5120=      DIMENSION X(100),Y(100),W(100,40),WL(40)
5130=      NP1=N+1
5140=      DATA SLASH,DOT,STAR,OH,BLANK/1H/,1H.,1H*,1HO," "/
5150=      XMAX=X(N+1)
5160=      YMAX=YMIN=0
5170=      DO 10 I=1,N
5180=      IF(Y(I).GT.YMAX)YMAX=Y(I)
5190=      IF(Y(I).LT.YMIN)YMIN=Y(I)
5200=   10 CONTINUE
5210=      YOVER=YMAX-Y(100)
5220=      DO 20 I=1,100
5230=      DO 20 J=1,40
5240=      W(I,J)=BLANK
5250=      IF(I.EQ.1)W(I,J)=SLASH
5260=      IF(J.EQ.20)W(I,J)=DOT
5270=      W(10,20)=W(20,20)=W(30,20)=W(40,20)=W(50,20)=W(99,20)="0"
5280=      W(60,20)=W(70,20)=W(80,20)=W(90,20)=W(100,20)="0"
5290=      W(9,20)=W(98,20)="1"
5300=      W(19,20)="2"
5310=      W(29,20)="3"
5320=      W(39,20)="4"
5330=      W(49,20)="5"
5340=      W(59,20)="6"
5350=      W(69,20)="7"
5360=      W(79,20)="8"
5370=      W(89,20)="9"
5380=   20 CONTINUE
```

```
      SCALE=YMAX/2Ø.
      STORE=-YMIN/2Ø.
      IF(SCALE.LT.STORE)SCALE=STORE
      DX=XMAX/5.
      SCALE=SCALE+.1*SCALE
      GMAX=2Ø.*SCALE
      DO 3Ø I=1,21
      XIM1=(I-1)
      STORE=XIM1*SCALE
      K=21-I
      L=19+I
       WL(L)=-STORE
       WL(K)=STORE
   3Ø CONTINUE
      DO 4Ø I=1,26,5
      J=4*(I-1)
      XI=(I-1)/5
      WL(I)=XI*DX
   4Ø CONTINUE
      DXX=XMAX/1ØØ.
      DO 5Ø I=1,N
      DO 44 K=1,N
       XK=K
       IF(X(I).GT.XK*DXX-DXX/2..AND.X(I).LE.XK*DXX+DXX/2.)GO TO 45
   44 CONTINUE
   45 CONTINUE
      DO 5Ø J=1,49
      XJM1=J-1
      XJ=J
      IF((GMAX-XJM1*SCALE.GT.Y(I)).AND.(GMAX-XJ*SCALE
     1.LE.Y(I)))W(K,J)=OH
   5Ø CONTINUE
      WRITE(2,1ØØ)XMAX,YMAX,YMIN,YOVER
      K=5
      WRITE(2,2ØØ)GMAX
      DO 6Ø J=1,4Ø
      IF(J.EQ.K)GO TO 55
      WRITE(2,4ØØ)(W(I,J),I=1,N)
      GO TO 6Ø
   55 K=K+5
      WRITE(2,2ØØ)WL(J),(W(I,J),I=1,N)
   6Ø CONTINUE
      WRITE(2,3ØØ)(WL(I),I=1,26,5)
  1ØØ FORMAT(///,T3Ø,"THIS PLOT HAS XMAX= ",1PE1Ø.3,",YMAX= "
     1,E1Ø.3,",YMIN= ",E1Ø.3," AND Y OVER= ",E1Ø.3,///)
  2ØØ FORMAT(5X,G1Ø.3,2X,1ØØA1)
  3ØØ FORMAT(///,T14,6(1PE1Ø.3,1ØX))
  4ØØ FORMAT(17X,1ØØA1)
      RETURN
      END
```

Appendix F

User's Guide for HUFF

This appendix describes the data card input, the scope control cards, and selecting the output when operating program HUFF. The data card input will be presented first followed by a discussion of the scope control cards. The last section will discuss the types of output available to the user.

Data Cards for HUFF

HUFF is programmed to solve two types of problems. The first type of problem is the nuclear blast in air and the second type involves the compression of solid materials with a constant applied pressure. Both problems require an input of three data cards. The first two data cards establish the parameters used in operating the programs and will be discussed first. The last card is the problem defining card and has a format based on the type of problem. All the data is read by HUFF with a list directed format allowing the user maximum flexibility in defining the input numbers.

Parameters. The two input cards defining the program parameters are as follows:

1. Card 1 - FLAG (2), FLAG (3), FLAG (4), TSTOP. This card contains the input flags defined as follows:

   FLAG (2) is an integer representing the number of result times listed on card 2.

   FLAG (3) is an integer representing the maximum number of time step iterations. The program will stop if it reaches this many iterations.

FLAG (4) is an integer representing the type of problem to be solved

1 for a problem in air

2 for a problem in a solid

TSTOP is the maximum problem time allowed. The program will stop if it reaches this time. Results for this time will be plotted in addition to those requested on Card 2.

2. Card 2 - TABLE (1), TABLE (2), ..., TABLE (FLAG(2)). This card lists the requested result times. A maximum of 10 times can be requested due to the dimension of the array TABLE.

Problem Defining Card. This is the third and last input card needed to define the problem. The blast problem will be discussed first followed by the compression problem.

1. Card 3 - YKT, SP, SD, ST is the problem defining card for the blast problem. The variables are defined as follows:

   YKT - Yield of the blast in kilotons.

   SP, SD, ST - Altitude scaling factors from Table I on page 15 of the text.

2. Card 3 - L, GAMMA, RHO, C, RADIUS, PRESS are the problem defining variables for the compression problem. These variables are defined as follows:

   L - Problem geometry factor

   1 - Rectangular geometry

   2 - Cylindrical geometry

   3 - Spherical geometry

GAMMA - Gruneisen ratio fcr the solid

RHO - Normal density of a solid

C - Normal density sound speed in the solid

RADIUS - Uncompressed radius of the solid

PRESS - Pressure applied in the problem by the
outer cell

The pressure in the compression problem can be applied throughout the problem solution by including the card (P(100)=PRESS) after ENTRY CRUSH in subroutine SQUEEZE. If this fortran card is not included, the pressure will be applied only in the initial conditions corresponding to an impulse pressure. A comment card is included at the proper location in subroutine SQUELZE to identify this option.

Scope Control Cards for HUFF

This section presents the scope control cards necessary to execute the HUFF program at AFIT. The sequence of control cards used will depend on what the user plans to do with the program and its results. A number of options will be discussed and explained to provide flexibility for the user.

Permanent File. To place HUFF on permanent file and execute a program simultaneously, the following control cards are necessary:

```
job,T200.problem
REQUEST,HUFF,*PF.
COPYBR,INPUT,HUFF.
REWIND,HUFF.
CATALOG,HUFF,RP=999.
REWIND,HUFF.
FTN,I=HUFF.
LGO.
7/8/9
HUFF DECK
7/8/9
DATA Cards for HUFF
7/8/9
6/7/8/9 (Orange Card)
```

After HUFF is on permanent file, the following control cards can be used:

```
job,T200.problem
ATTACH,HUFF.
FTN,I=HUFF.
LGO.
7/8/9
DATA Cards for HUFF
7/8/9
6/7/8/9 (Orange Card)
```

Single Operation. The following cards use HUFF to solve a single problem without using a permanent file:

```
job,T200,problem
FTN.
LGO.
7/8/9
HUFF PROGRAM
7/8/9
DATA for HUFF
7/8/9
6/7/8/9 (Orange Card)
```

Listing Local Files. HUFF writes all the plotted information and summary information on local files called FILES and REVIEW respectively. The following cards will route a listing of the local files to the printer:

```
job,T200.problem
FTN.
LGO.
REWIND,FILES.
ROUTE,FILES,TID=BB,DC=PR,FID=job.
REWIND,REVIEW.
ROUTE,REVIEW,TID=BB,DC=PR,FID=job.
7/8/9
HUFF PROGRAM
7/8/9
DATA
7/8/9
6/7/8/9 (Orange Card)
```

Save Local Files. To save local files on a permanent file, the following cards are necessary:

```
job,T200.problem
Request,lfn,*PF.
FTN.
LGO.
REWIND,lfn.
CATALOG,lfn,RP=999.
7/8/9
HUFF PROGRAM
7/8/9
DATA for HUFF
7/8/9
6/7/8/9 (Orange Card)
```

There are many more combinations available in using scope cards to run a program. The above listings should provide sufficient examples to operate the program.

HUFF Output. A large amount of useful information is generated during a problem solution. This information is easily analyzed if it is presented in graphic form. However, if a more detailed analysis of the results is required, a tabular output may be preferred. For this reason, a variety of outputs are generated by HUFF. A tabular form of all the results is placed on a local file called FILES, a summary of the parameters of the problem is placed on a local file called REVIEW and all the information is plotted on the output file by subroutine PLOTSM.

Plot routines using DISSPLA or CALCOMP can also be used with HUFF by replacing subroutine PLOTSM with the desired plot routine. If FILES and REVIEW are placed on a permanent file, selected data can be read from these files and plotted, after examing the results displayed on the output file, without reexecuting the entire program.

Vita

David J. Peters was born on 23 December 1948 in Kenosha, Wisconsin. He graduated from Tremper High School in Kenosha, Wisconsin, in 1966. He attended Wisconsin State University at Whitewater graduating May 1970 with a degree of Bachelor of Science in Mathematics and Physics. In November 1970, he entered the Air Force receiving his commission from Officer Training School in February 1971. He then attended Undergraduate Pilot Training at Webb AFB, Texas, receiving his aviation wings in February 1972. After pilot training, he completed transition training for the C-141 aircraft at Altus and was assigned to the 8th Military Airlift Squadron, McChord AFB, Washington as a C-141 pilot. While stationed at McChord AFB, he pursued a Masters Degree in Business Administration sponsored by Southern Illinois University - Edwardsville and graduated with a MBA in December 1975. In January 1976, he was assigned to Air Training Command through a MAC exchange program. He completed the T-38 Pilot Instructor Training program at Randolph AFB, Texas, in July 1976 and was assigned as a T-38 Instructor Pilot in the 50th flying training squadron at Columbus AFB, Mississippi. He performed as an Instructor Pilot until entering the School of Engineering, Air Force Institute of Technology in June 1977.

Permanent Address: 2600 Fasske Street
Sulphur, Louisiana 70663

(14) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. ~~REPORT NUMBER~~
AFIT/GNE/PH/78D-9

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE ~~(and Subtitle)~~
(16) HUFF, A ONE-DIMENSIONAL HYDRODYNAMICS CODE FOR STRONG SHOCKS,

5. ~~TYPE OF REPORT & PERIOD CO~~VERED
(9) Master's [redacted] Thesis,

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) David J. Peters
Captain USAF

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

12. REPORT DATE
(11) Dec[redacted]78

13. NUMBER OF PAGES
~~95~~ (12) 107p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
Approved for public release; IAW AFR 190-17
JOSEPH P. HIPPS, MAJOR, USAF
Director of Information
1-23-79

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

HUFF is a one-dimensional Lagrangian hydrodynamics computer code developed from the basic principles of mass, momentum, and energy conservation for strong shock propagation in a solid or gas. Two equations of state are used - the adiabatic ideal gas law with a variable gamma and the Gruneisen solid equation of state with a constant Gruneisen ratio. The Richtmyer and Morton difference equations for strong shocks are used on a spatial mesh composed of up to 100 cells. Results for two problems are presented which show → over

012225

the usefulness and limitations of the code and also serve as sample problems. The results of a one kiloton nuclear explosion are compared to the Nuclear Blast Standard 1KT. The results were within 13 percent for shock overpressure and overdensity, 5 percent for shock material velocity, and 2 percent for shock position over a range of 20 meters to 2 kilometers from the burst point. The larger deviations occurred at early times being attributed to an absence of radiation transport calculations in the code. The second problem, a megabar compression of uranium, shows agreement within two percent for all parameters (peak shock pressure, density, material velocity and shock velocity) when compared with the Rankine-Hugoniot compression curves. The equation of state for a solid was limited to calculations below 100 megabars due to its simplicity and constant value for the Gruneisen ratio. A complete users guide and program listing are also provided.